Project management system
Software Requirements Specification
Version <1.0>

Revision History

Date	Version	Description	Author
<04/13/07>	<1.0>	SRS 1.0	Group-2

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose:

The purpose of the project is to maintain the details of projects, employees and admin. The main purpose of this project is to maintain an easy circulation system between admin and the employees, resources allocated to the project, close the project, also to search the employees for project allocation and to maintain details about the admin, employee and project. (Id, password, name).

1.2 Scope:

The Project Management System addresses the management of software projects. It provides the framework for organizing and managing resources in such a way that these resources deliver all the work required to complete a software project within defined scope, time and cost constraints.

The e Integrated Project Management System provides information's like details of the projects, resources allocated to new projects, close projects.

1.3 Overview:

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 3 gives the functional requirements, data requirements. It also gives the admin viewpoint of project. Section 3 also gives the specific requirements of the project. Section 3 also discusses the external interface requirements and gives detailed description of functional requirements. Section 4 is for supporting information.

2. Overall Description:

An Employee project management system is software that is designed to manage Resources allocation to project and deallocation. It helps admin to maintain the database of new project, current running projects and closed projects. This system completely automates all your projects activities.

The SRS will provide a detailed description of the Employee project management system. This document will provide the outline of the requirements, overview of the characteristics and constraints of the system. This section of the SRS will provide the general factors that affect the product and its requirements.

3. Specific Requirements

The specific requirements are –

3.1 Functionality:

3.1.1 Admin:

3.1.1.1 Add admin:

- Input: Enter the details of the admin such as names ,password ,Id
- Output: confirmation of admin adding
- Processing: All details will be checked and if any error are found then an error message is displayed else admin name ,password ,Id will be generated

3.1.1.2 Admin log-in:

- Input: Enter the admin id and password
- Output: Admin will be able to use the features of software.

3.1.1.3 Resources allocated to project:

- State: Searched the project and employee admin wants to Resources allocation
- Input: Click the project and employee admin wants
- Output: conformation for Resources allocation.
- Processing: if selected project and employee is available Resources allocated then else error will be displayed.

3.1.1.4 Close project:

- State: Searched the project admin wants to Resources deallocation
- Input: Click the project admin wants
- Output: conformation for Resources deallocation.
- Processing: if selected project is available Resources deallocated then else error will be displayed.

3.1.2 Project:

3.1.2.1 Add project:

- Input: Enter the details of the project such as name, start date, end date, no of resources required.
- Output: confirmation of adding the project.

3.1.2.2 Search project:

- Input: Searched the projects admin wants

- Output: List of projects
- Processing: if projects are available display the projects then else error will be displayed

3.1.2.3 Update project:

- State: Searched the project admin wants to update.
- Input: click the project admin wants.
- Output: conformation for project update.
- Processing: if selected project is available then project will be update else error will be displayed.

3.1.2.4 Edit project:

- State: Searched the project admin wants to edit.
- Input: click the project admin wants.
- Output: conformation for project edit.
- Processing: if selected project is available then project will be edit else error will be displayed.

3.1.2.5 Remove project:

- State: Searched the project admin wants to remove.
- Input: click the project admin wants.
- Output: Conformations for project remove.
- Processing: if selected project is available then project will be removing else error will be displayed.

3.1.3 Employee:

3.1.3.1 Add employee:

- Input: Enter the details of the employee such as name, password, gander, basic salary required.
- Output: confirmation of adding the project.

3.1.3.2 Search employee:

- Input: Searched the employees admin wants
- Output: List of employee.

- Processing: if employees are available display the employees then else error will be displayed.

3.1.3.3 Update employee:

- State: Searched the employee admin wants to update.
- Input: click the employee admin wants.
- Output: conformation for employee update.
- Processing: if selected employee is available then employee will be update else error will be displayed.

3.1.3.4 Edit employee:

- State: employee log-in
- Input: click the edit employee name or password.
- Output: conformation for employee edit.
- Processing: if selected employee is available then employee will be edit else error will be displayed.

3.1.3.5 Remove employee:

- State: Searched the employee admin wants to remove.
- Input: click the employee for remove.
- Output: Conformation for employee removes.
- Processing: if selected employee is available then employee will be removing else error will be displayed.

3.1.3.6 Employee log-in:

- Input: Enter the employee id and password
- Output: Employee will be able to use the features of software.

3.2 USER CHARACTERSTICS

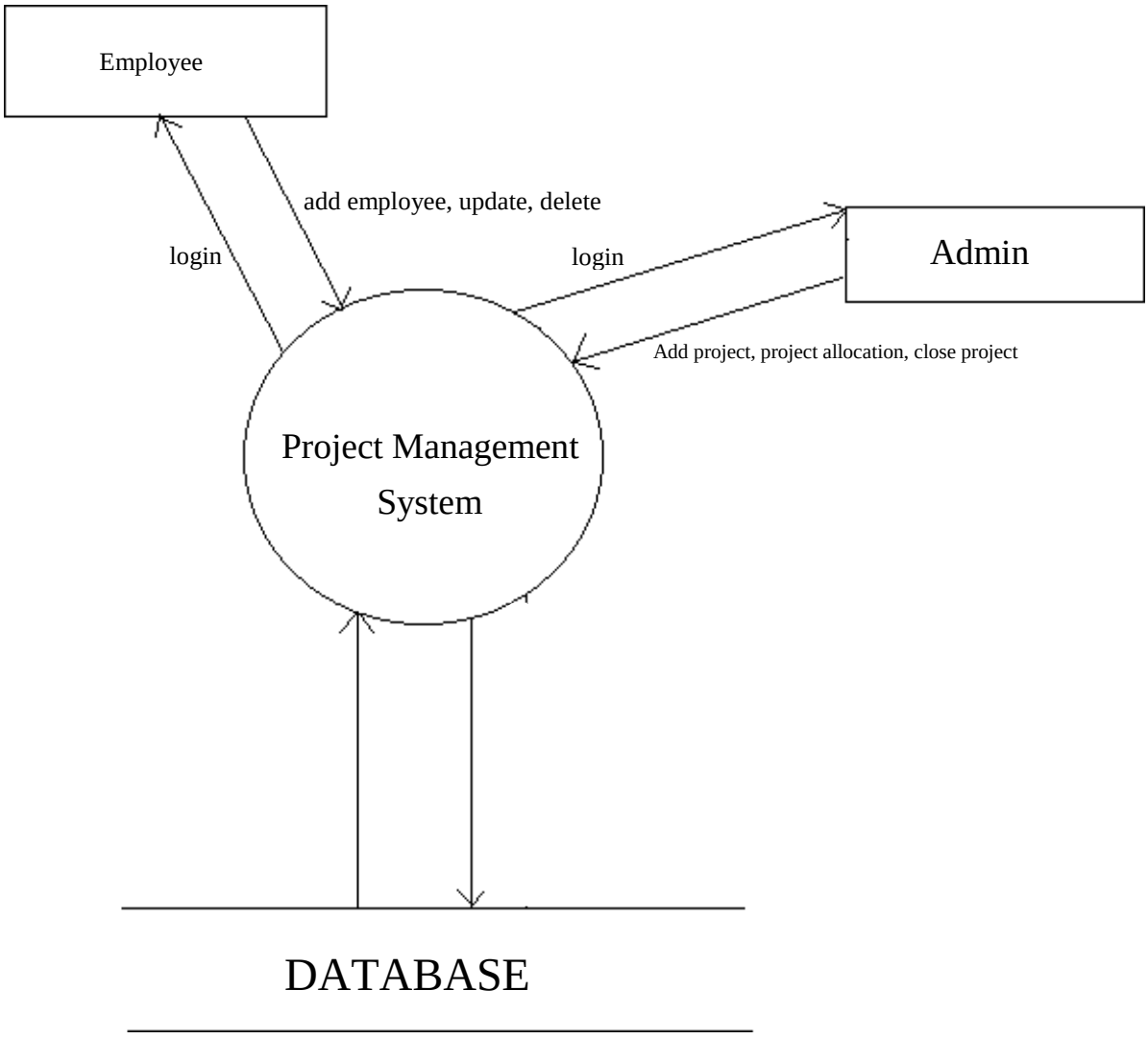
We have 2 levels of users:

- Admin:

- Add new project
- Update project
- Resource allocated for project
- Close project

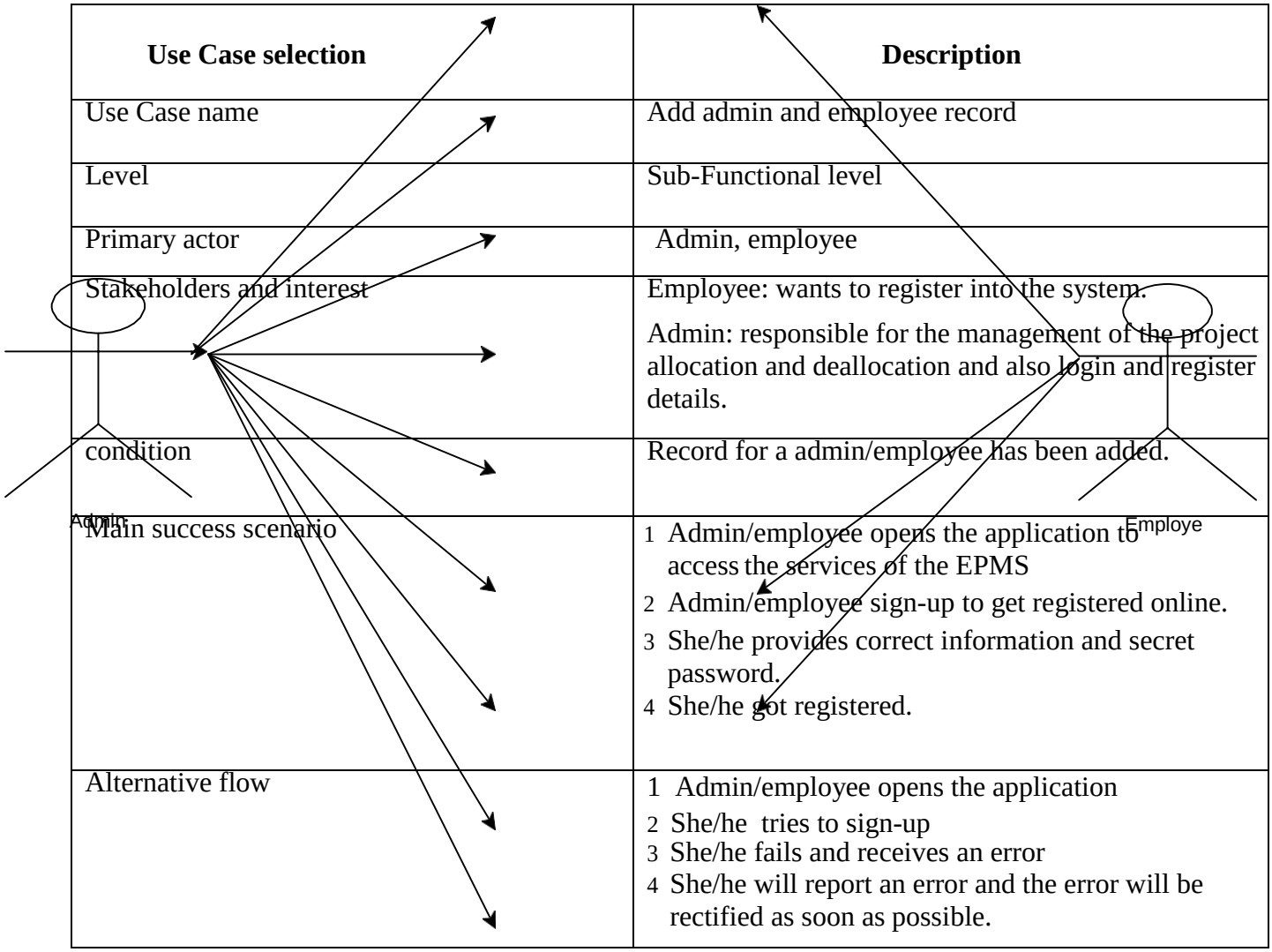
- Employee:
  - Add new employee
  - Remove employee
  - Edit employee details
  - Read employee details

3.3 FLOW DIAGRAM

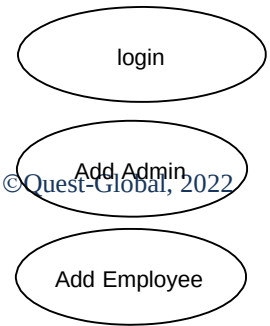


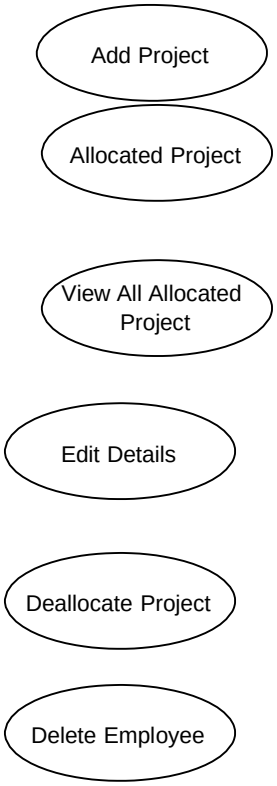
3.4 USE CASE MODEL DESCRIPTION

Use Case selection	Description
Use Case name	Add admin and employee record
Level	Sub-Functional level
Primary actor	Admin, employee
Stakeholders and interest	Employee: wants to register into the system. Admin: responsible for the management of the project allocation and deallocation and also login and register details.
condition	Record for a admin/employee has been added.
Main success scenario	1 Admin/employee opens the application to access the services of the EPMS 2 Admin/employee sign-up to get registered online. 3 She/he provides correct information and secret password. 4 She/he got registered.
Alternative flow	1 Admin/employee opens the application 2 She/he tries to sign-up 3 She/he fails and receives an error 4 She/he will report an error and the error will be rectified as soon as possible.

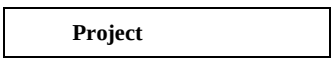


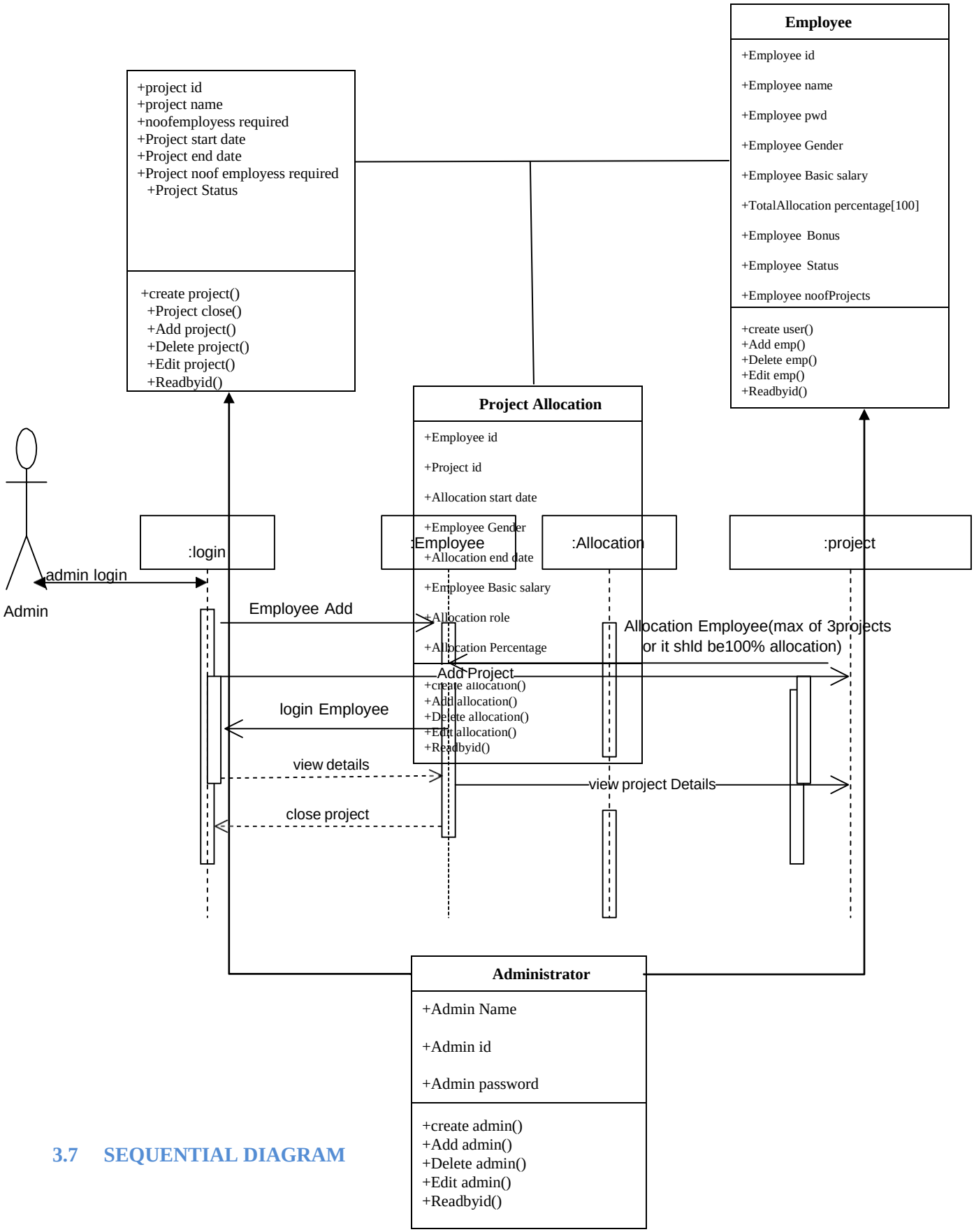
3.4 USE CASE DIAGRAM





3.6 CLASS DIAGRAM





3.7 SEQUENTIAL DIAGRAM

4. Supporting Information

Please refer the following document:

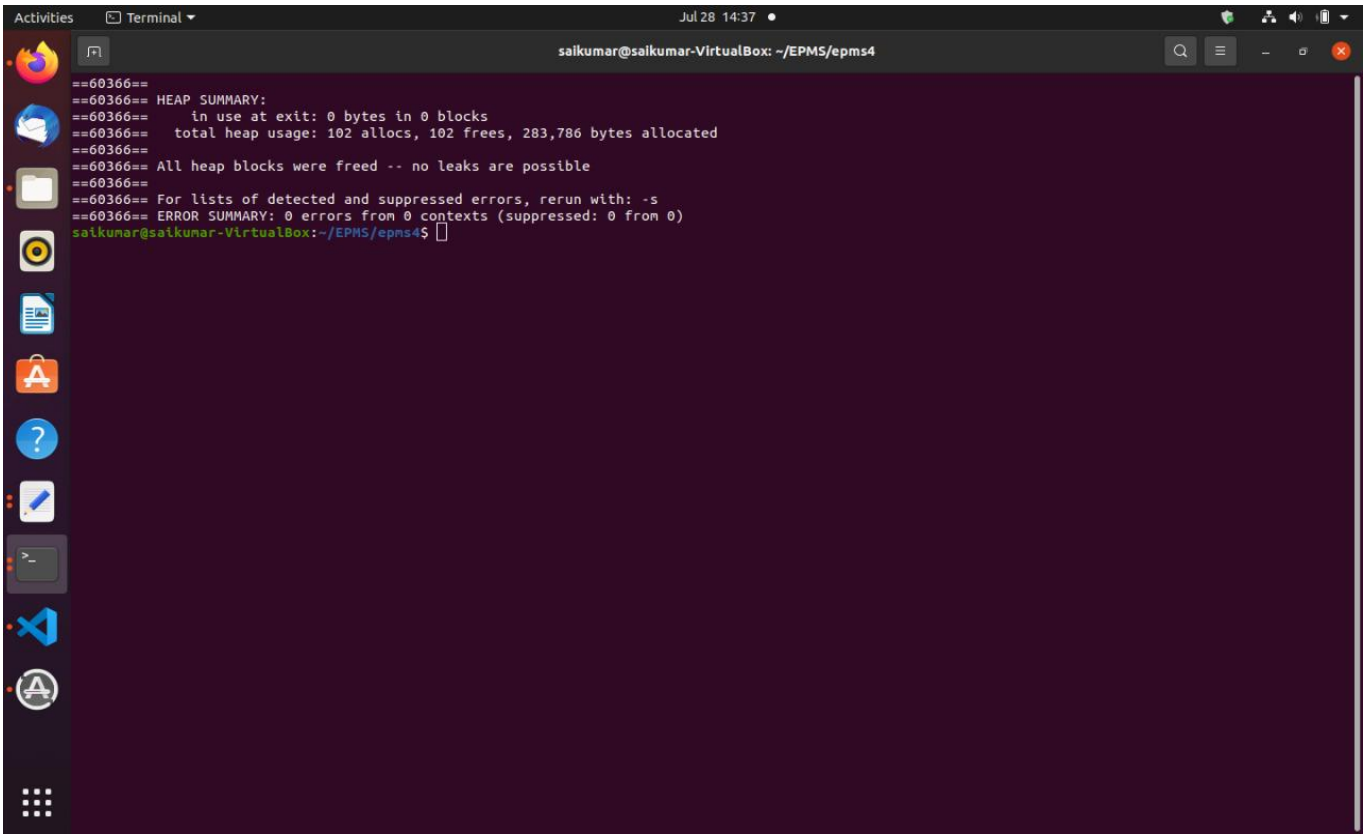
1. Project management system ppt.

2. High level design document.
3. Low level design document.
4. Test case document.

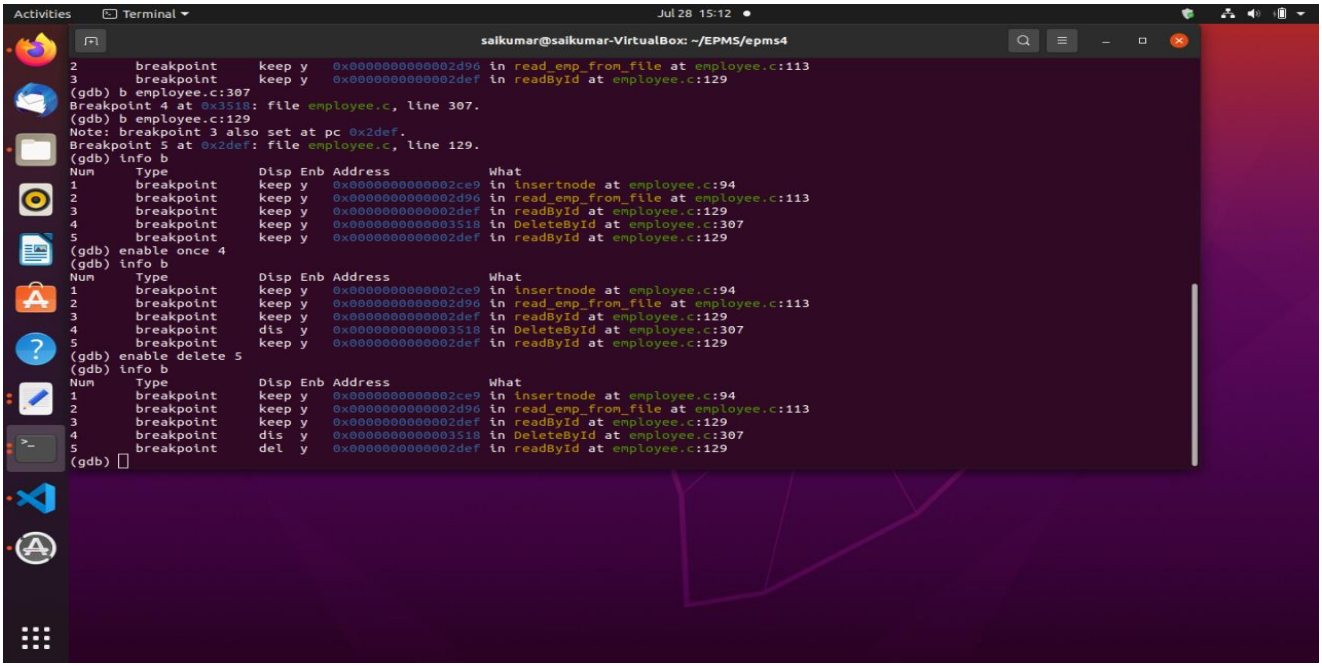
5. Conclusion

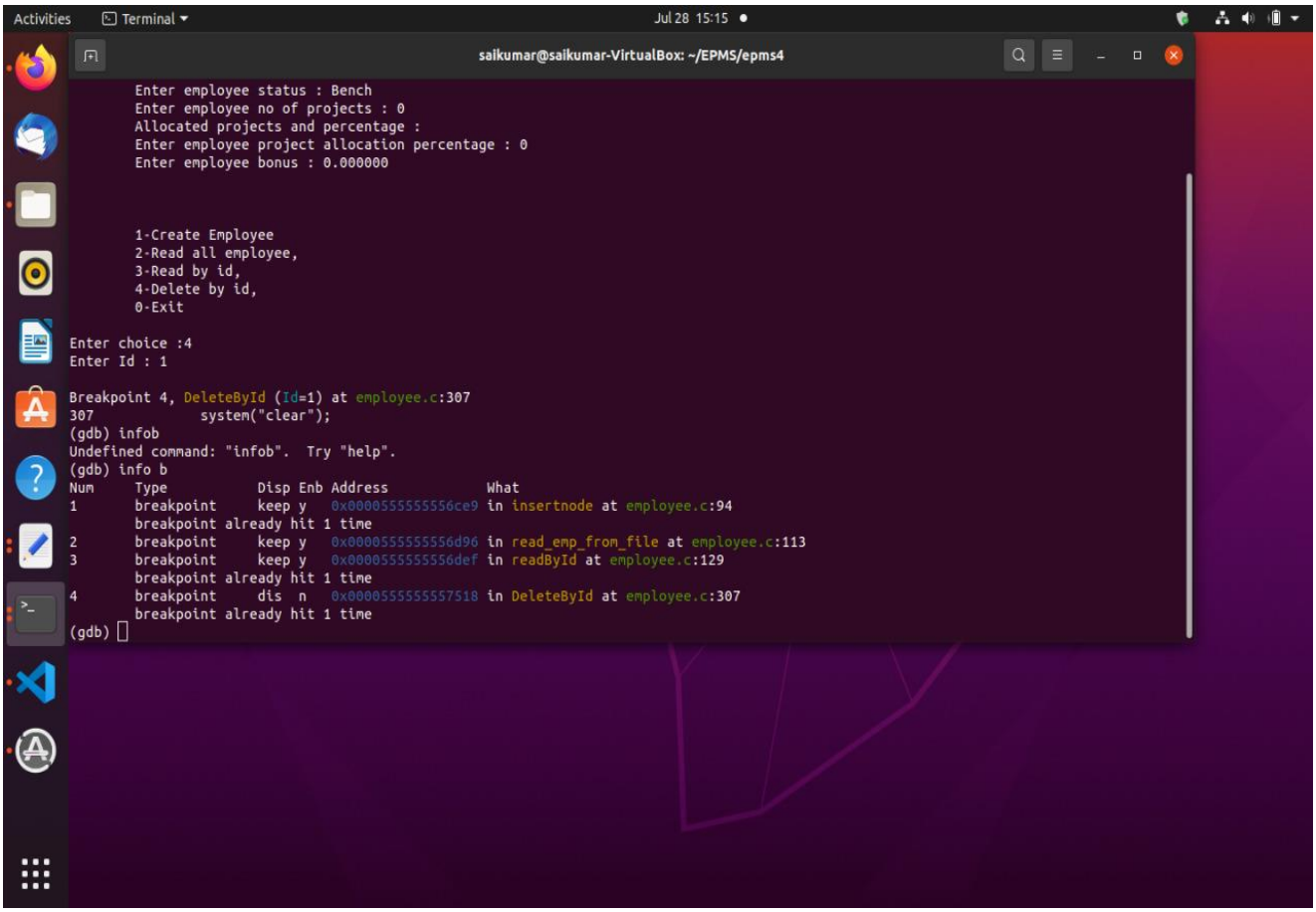
The application created and tested all possible cases successful .The purpose of an employee project management system is to manage & track the daily work of the project such as add project, resources allocated to project resources deallocated to project, etc.

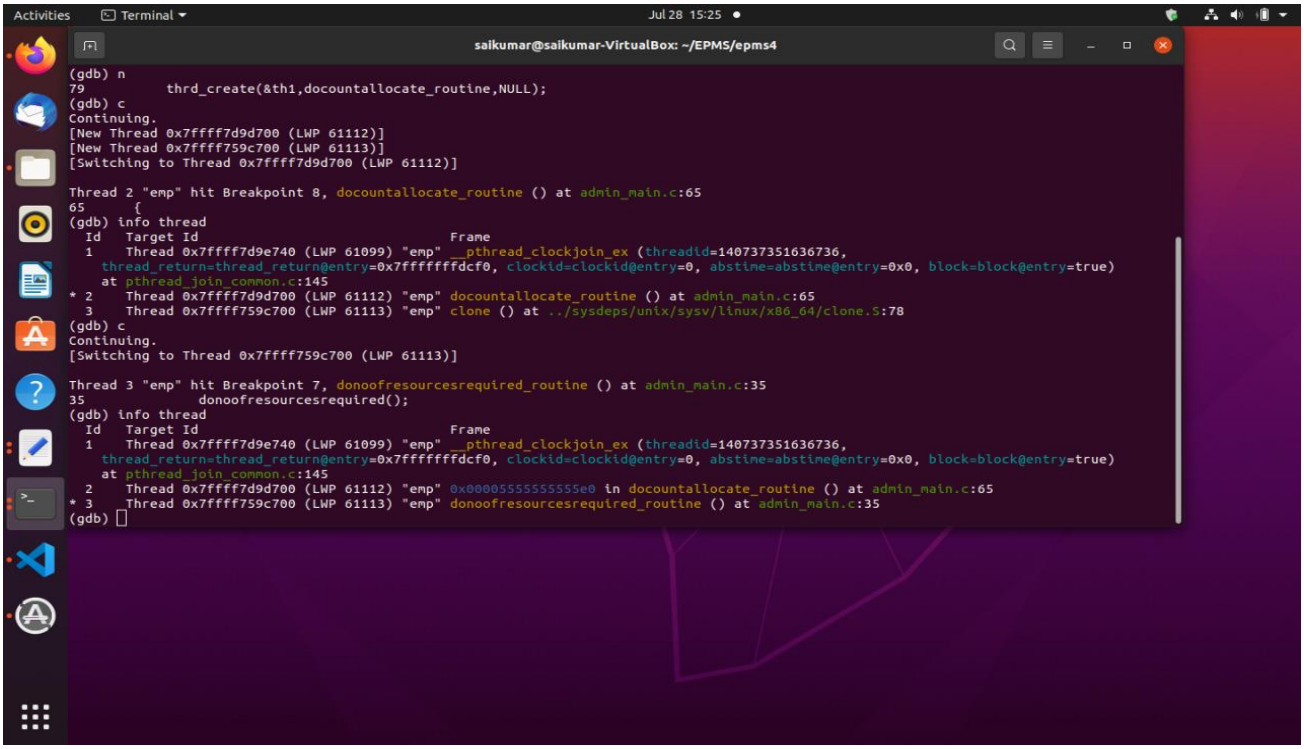
6. valgrind-memory



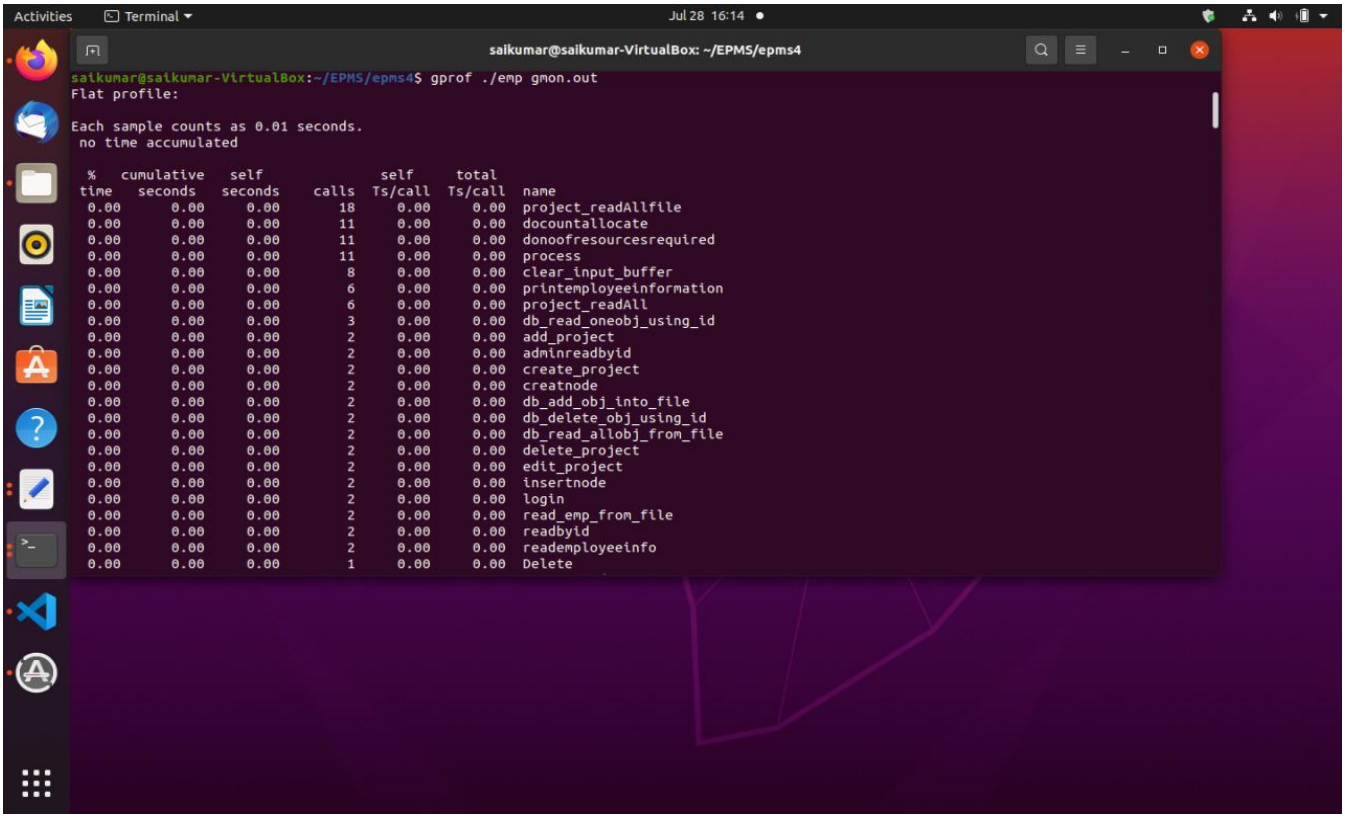
7. Gdb compilation

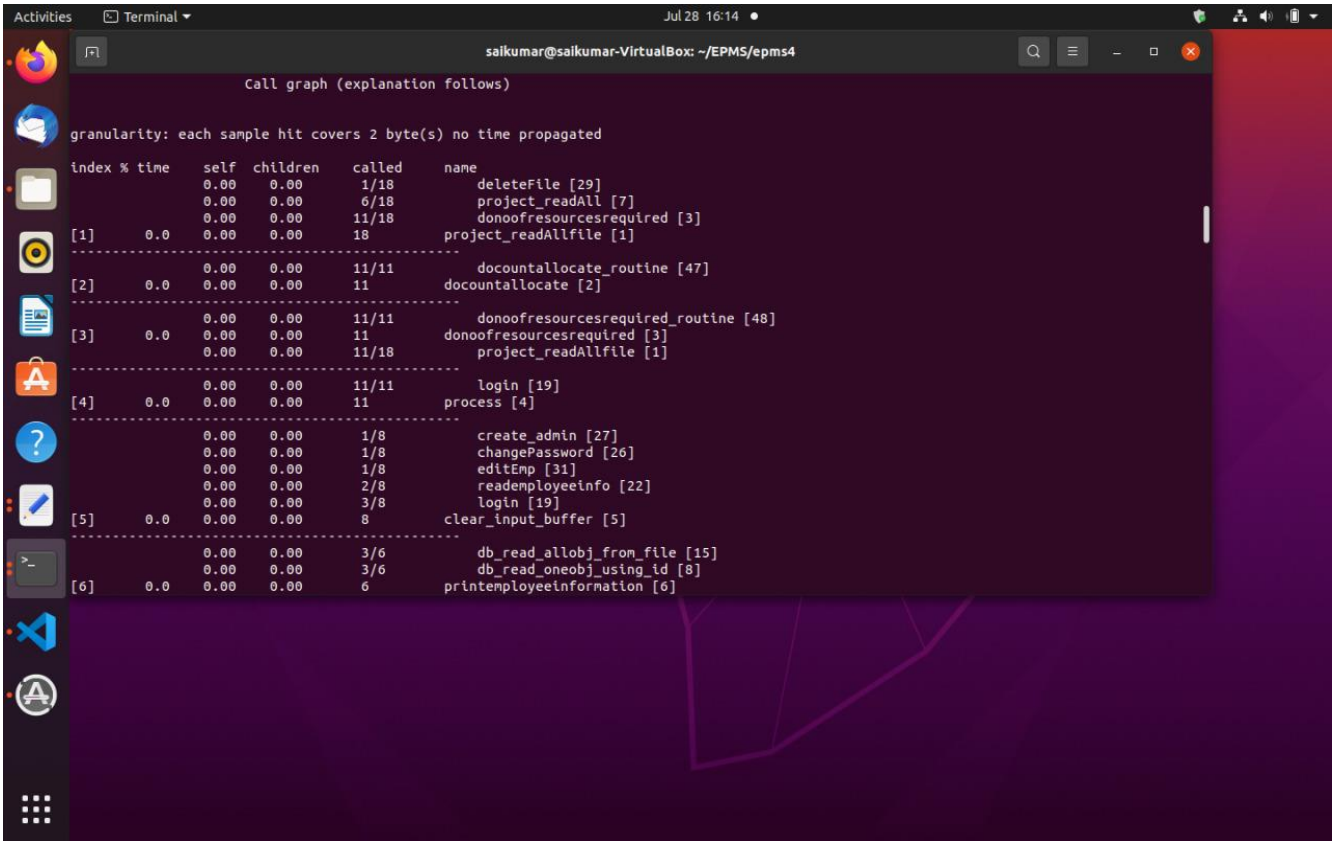

Activities
Terminal
Jul 28 15:12
saikumar@saikumar-VirtualBox: ~/EPMS/epms4
2 breakpoint keep y 0x0000000000002d96 in read_emp_from_file at employee.c:113
3 breakpoint keep y 0x0000000000002def in readById at employee.c:129
(gdb) b employee.c:307
Breakpoint 4 at 0x3518: file employee.c, line 307.
(gdb) b employee.c:129
Note: breakpoint 3 also set at pc 0x2def.
Breakpoint 5 at 0x2def: file employee.c, line 129.
(gdb) info b
Num Type Disp Enb Address What
1 breakpoint keep y 0x0000000000002ce9 in insertnode at employee.c:94
2 breakpoint keep y 0x0000000000002d96 in read_emp_from_file at employee.c:113
3 breakpoint keep y 0x0000000000002def in readById at employee.c:129
4 breakpoint keep y 0x0000000000003518 in DeleteById at employee.c:307
5 breakpoint keep y 0x0000000000002def in readById at employee.c:129
(gdb) enable once 4
(gdb) info b
Num Type Disp Enb Address What
1 breakpoint keep y 0x0000000000002ce9 in insertnode at employee.c:94
2 breakpoint keep y 0x0000000000002d96 in read_emp_from_file at employee.c:113
3 breakpoint keep y 0x0000000000002def in readById at employee.c:129
4 breakpoint dis y 0x0000000000003518 in DeleteById at employee.c:307
5 breakpoint keep y 0x0000000000002def in readById at employee.c:129
(gdb) enable delete 5
(gdb) info b
Num Type Disp Enb Address What
1 breakpoint keep y 0x0000000000002ce9 in insertnode at employee.c:94
2 breakpoint keep y 0x0000000000002d96 in read_emp_from_file at employee.c:113
3 breakpoint keep y 0x0000000000002def in readById at employee.c:129
4 breakpoint dis y 0x0000000000003518 in DeleteById at employee.c:307
5 breakpoint del y 0x0000000000002def in readById at employee.c:129
(gdb)

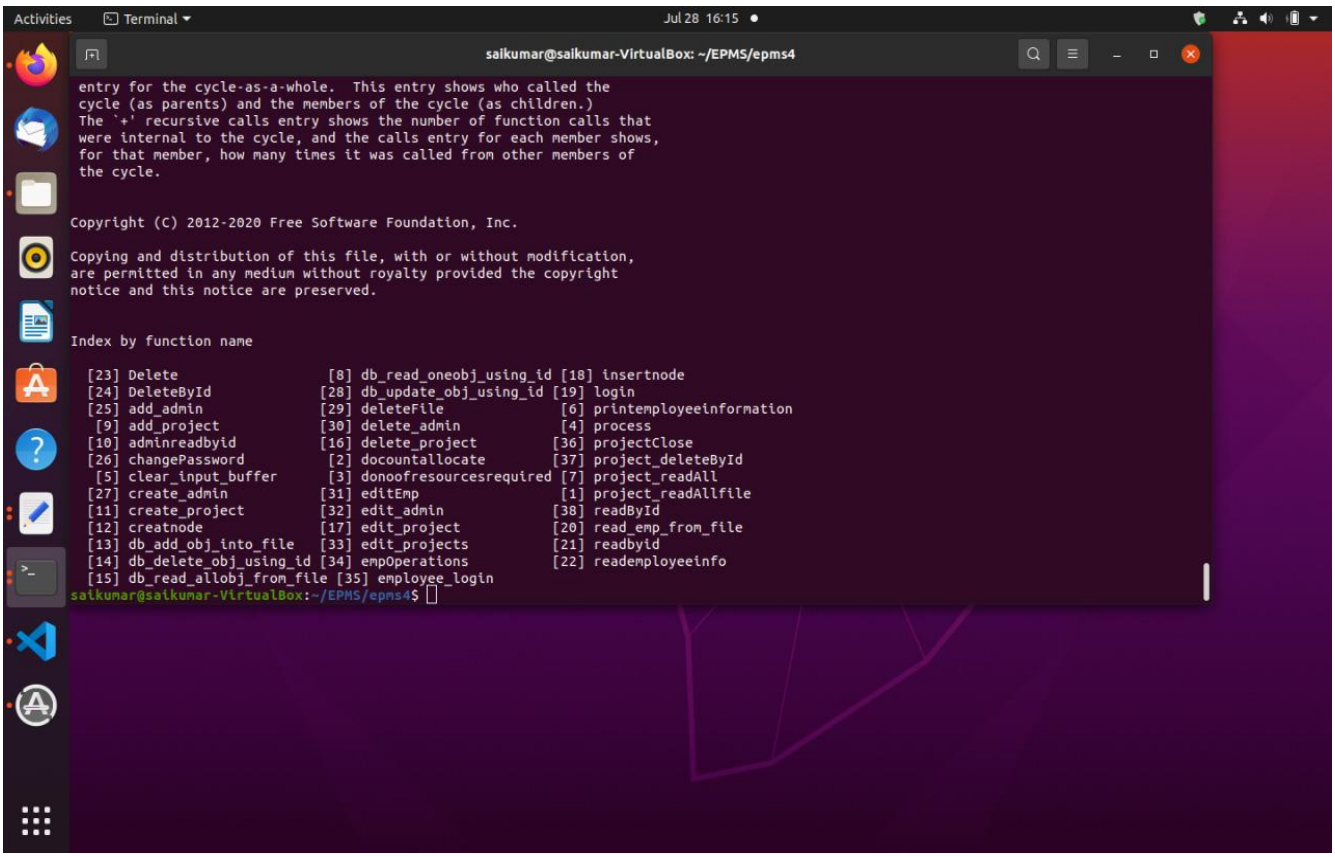

Activities
Terminal
Jul 28 15:15
saikumar@saikumar-VirtualBox: ~/EPMS/epms4
Enter employee status : Bench
Enter employee no of projects : 0
Allocated projects and percentage :
Enter employee project allocation percentage : 0
Enter employee bonus : 0.000000
1-Create Employee
2-Read all employee,
3-Read by id,
4-Delete by id,
0-Exit
Enter choice :4
Enter Id : 1
Breakpoint 4, DeleteById (Id=1) at employee.c:307
307 system("clear");
(gdb) infob
Undefined command: "infob". Try "help".
(gdb) info b
Num Type Disp Enb Address What
1 breakpoint keep y 0x00005555555556ce9 in insertnode at employee.c:94
breakpoint already hit 1 time
2 breakpoint keep y 0x00005555555556d96 in read_emp_from_file at employee.c:113
3 breakpoint keep y 0x00005555555556def in readById at employee.c:129
breakpoint already hit 1 time
4 breakpoint dis n 0x0000555555557518 in DeleteById at employee.c:307
breakpoint already hit 1 time
(gdb)


Activities
Terminal
Jul 28 15:25
saikumar@saikumar-VirtualBox: ~/EPMS/epms4
(gdb) n
79 thrd_create(&th1,docountallocate_routine,NULL);
(gdb) c
Continuing.
[New Thread 0x7ffff7d9d700 (LWP 61112)]
[New Thread 0x7ffff759c700 (LWP 61113)]
[Switching to Thread 0x7ffff7d9d700 (LWP 61112)]
Thread 2 "emp" hit Breakpoint 8, docountallocate_routine () at admin_main.c:65
65 {
(gdb) info thread
Id Target Id Frame
1 Thread 0x7ffff7d9e740 (LWP 61099) "emp" __pthread_clockjoin_ex (threadid=140737351636736,
thread_return=thread_return@entry=0x7fffffffdcf0, clockid=clockid@entry=0, abstime=abstime@entry=0x0, block=block@entry=true)
at pthread_join_common.c:145
* 2 Thread 0x7ffff7d9d700 (LWP 61112) "emp" docountallocate_routine () at admin_main.c:65
3 Thread 0x7ffff759c700 (LWP 61113) "emp" clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:78
(gdb) c
Continuing.
[Switching to Thread 0x7ffff759c700 (LWP 61113)]
Thread 3 "emp" hit Breakpoint 7, donoofresourcesrequired_routine () at admin_main.c:35
35 donoofresourcesrequired();
(gdb) info thread
Id Target Id Frame
1 Thread 0x7ffff7d9e740 (LWP 61099) "emp" __pthread_clockjoin_ex (threadid=140737351636736,
thread_return=thread_return@entry=0x7fffffffdcf0, clockid=clockid@entry=0, abstime=abstime@entry=0x0, block=block@entry=true)
at pthread_join_common.c:145
2 Thread 0x7ffff7d9d700 (LWP 61112) "emp" 0x00005555555555e0 in docountallocate_routine () at admin_main.c:65
* 3 Thread 0x7ffff759c700 (LWP 61113) "emp" donoofresourcesrequired_routine () at admin_main.c:35
(gdb)

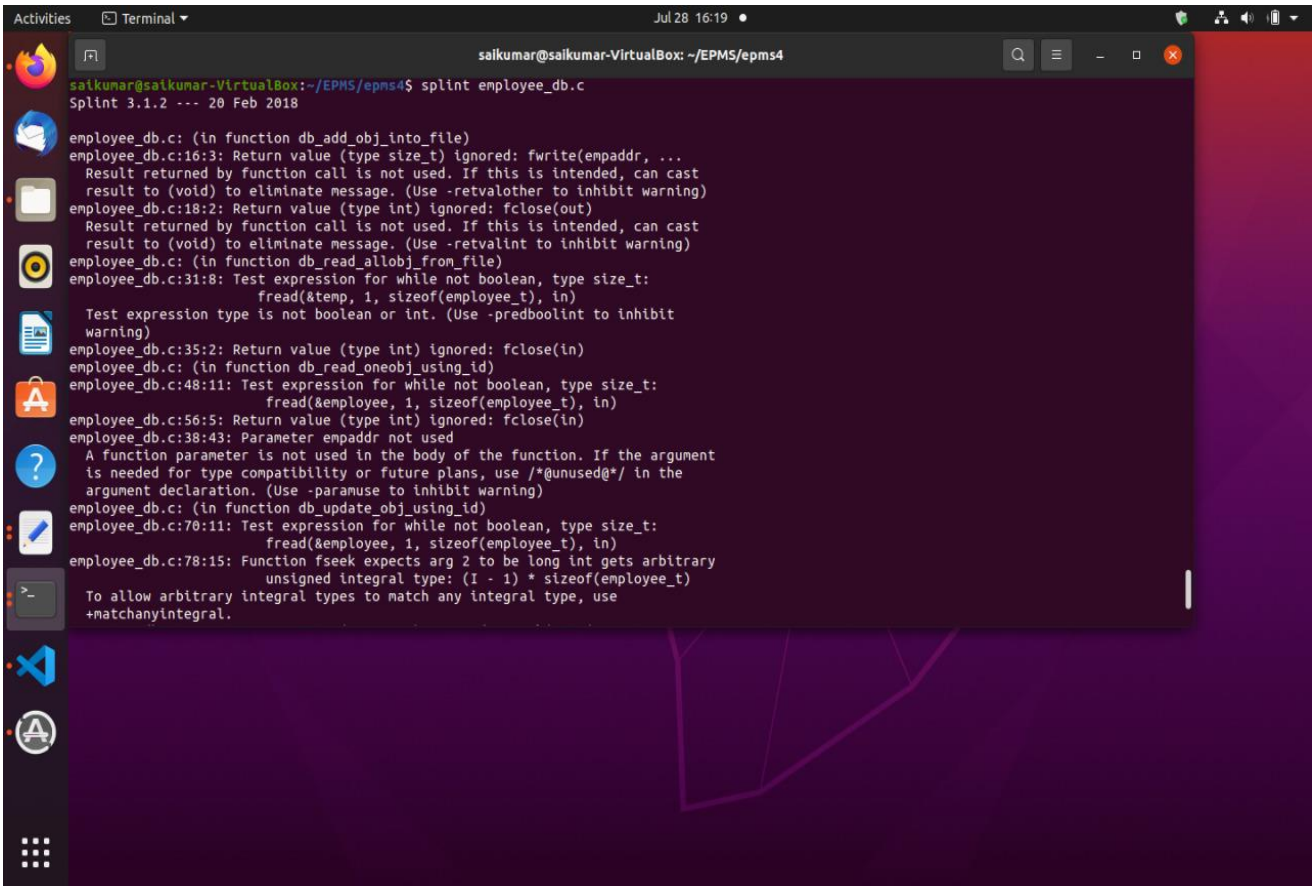
8. G profile


Activities
Terminal
Jul 28 16:14
saikumar@saikumar-VirtualBox: ~/EPMS/epms4
saikumar@saikumar-VirtualBox:~/EPMS/epms4$ gprof ./emp gmon.out
Flat profile:
Each sample counts as 0.01 seconds.
no time accumulated
% cumulative self self total
time seconds seconds calls Ts/call Ts/call name
0.00 0.00 0.00 18 0.00 0.00 project_readAllfile
0.00 0.00 0.00 11 0.00 0.00 docountallocate
0.00 0.00 0.00 11 0.00 0.00 donoofresourcesrequired
0.00 0.00 0.00 11 0.00 0.00 process
0.00 0.00 0.00 8 0.00 0.00 clear_input_buffer
0.00 0.00 0.00 6 0.00 0.00 printemployeeinformation
0.00 0.00 0.00 6 0.00 0.00 project_readAll
0.00 0.00 0.00 3 0.00 0.00 db_read_oneobj_using_id
0.00 0.00 0.00 2 0.00 0.00 add_project
0.00 0.00 0.00 2 0.00 0.00 adminreadbyid
0.00 0.00 0.00 2 0.00 0.00 create_project
0.00 0.00 0.00 2 0.00 0.00 creatnode
0.00 0.00 0.00 2 0.00 0.00 db_add_obj_into_file
0.00 0.00 0.00 2 0.00 0.00 db_delete_obj_using_id
0.00 0.00 0.00 2 0.00 0.00 db_read_allobj_from_file
0.00 0.00 0.00 2 0.00 0.00 delete_project
0.00 0.00 0.00 2 0.00 0.00 edit_project
0.00 0.00 0.00 2 0.00 0.00 insertnode
0.00 0.00 0.00 2 0.00 0.00 login
0.00 0.00 0.00 2 0.00 0.00 read_emp_from_file
0.00 0.00 0.00 2 0.00 0.00 readbyid
0.00 0.00 0.00 2 0.00 0.00 reademployeeinfo
0.00 0.00 0.00 1 0.00 0.00 Delete


Activities
Terminal
Jul 28 16:14
saikumar@saikumar-VirtualBox: ~/EPMS/epms4
Call graph (explanation follows)
granularity: each sample hit covers 2 byte(s) no time propagated
index % time self children called name
0.00 0.00 1/18 deleteFile [29]
0.00 0.00 6/18 project_readAll [7]
0.00 0.00 11/18 donoofresourcesrequired [3]
[1] 0.0 0.00 0.00 18 project_readAllfile [1]
0.00 0.00 11/11 docountallocate_routine [47]
[2] 0.0 0.00 0.00 11 docountallocate [2]
0.00 0.00 11/11 donoofresourcesrequired_routine [48]
[3] 0.0 0.00 0.00 11 donoofresourcesrequired [3]
0.00 0.00 11/18 project_readAllfile [1]
0.00 0.00 11/11 login [19]
[4] 0.0 0.00 0.00 11 process [4]
0.00 0.00 1/8 create_admin [27]
0.00 0.00 1/8 changePassword [26]
0.00 0.00 1/8 editEmp [31]
0.00 0.00 2/8 reademployeeinfo [22]
0.00 0.00 3/8 login [19]
[5] 0.0 0.00 0.00 8 clear_input_buffer [5]
0.00 0.00 3/6 db_read_allobj_from_file [15]
0.00 0.00 3/6 db_read_oneobj_using_id [8]
[6] 0.0 0.00 0.00 6 printemployeeinformation [6]


Activities
Terminal
Jul 28 16:15
saikumar@saikumar-VirtualBox: ~/EPMS/epms4
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.
Copyright (C) 2012-2020 Free Software Foundation, Inc.
Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.
Index by function name
[23] Delete [8] db_read_oneobj_using_id [18] insertnode
[24] DeleteById [28] db_update_obj_using_id [19] login
[25] add_admin [29] deleteFile [6] printemployeeinformation
[9] add_project [30] delete_admin [4] process
[10] adminreadbyid [16] delete_project [36] projectClose
[26] changePassword [2] docountallocate [37] project_deleteById
[5] clear_input_buffer [3] donoofresourcesrequired [7] project_readAll
[27] create_admin [31] editEmp [1] project_readAllfile
[11] create_project [32] edit_admin [38] readById
[12] creatnode [17] edit_project [20] read_emp_from_file
[13] db_add_obj_into_file [33] edit_projects [21] readbyid
[14] db_delete_obj_using_id [34] empOperations [22] reademployeeinfo
[15] db_read_allobj_from_file [35] employee_login
saikumar@saikumar-VirtualBox:~/EPMS/epms4$

9. Splint


Activities
Terminal
Jul 28 16:19
saikumar@saikumar-VirtualBox: ~/EPMS/epms4
saikumar@saikumar-VirtualBox:~/EPMS/epms4$ splint employee_db.c
Splint 3.1.2 --- 20 Feb 2018
employee_db.c: (in function db_add_obj_into_file)
employee_db.c:16:3: Return value (type size_t) ignored: fwrite(empaddr, ...
Result returned by function call is not used. If this is intended, can cast
result to (void) to eliminate message. (Use -retvalother to inhibit warning)
employee_db.c:18:2: Return value (type int) ignored: fclose(out)
Result returned by function call is not used. If this is intended, can cast
result to (void) to eliminate message. (Use -retvalint to inhibit warning)
employee_db.c: (in function db_read_allobj_from_file)
employee_db.c:31:8: Test expression for while not boolean, type size_t:
fread(&temp, 1, sizeof(employee_t), in)
Test expression type is not boolean or int. (Use -predboolint to inhibit
warning)
employee_db.c:35:2: Return value (type int) ignored: fclose(in)
employee_db.c: (in function db_read_oneobj_using_id)
employee_db.c:48:11: Test expression for while not boolean, type size_t:
fread(&employee, 1, sizeof(employee_t), in)
employee_db.c:56:5: Return value (type int) ignored: fclose(in)
employee_db.c:38:43: Parameter empaddr not used
A function parameter is not used in the body of the function. If the argument
is needed for type compatibility or future plans, use /*@unused@*/ in the
argument declaration. (Use -paramuse to inhibit warning)
employee_db.c: (in function db_update_obj_using_id)
employee_db.c:70:11: Test expression for while not boolean, type size_t:
fread(&employee, 1, sizeof(employee_t), in)
employee_db.c:78:15: Function fseek expects arg 2 to be long int gets arbitrary
unsigned integral type: (I - 1) * sizeof(employee_t)
To allow arbitrary integral types to match any integral type, use
+matchanyintegral.