

BUG TRACKING SYSTEM

Design

**Index**

1. Introduction ------------------------------------------------ 3

1.1 Intended audience ------------------------------------------------ 3

1.2 Project purpose ------------------------------------------------ 3

1.3 Key project objective ------------------------------------------------ 3

1.4 Project scope and limitation ------------------------------------------------ 3

1.5 Functional overview ------------------------------------------------ 3

1.5.1Header files ------------------------------------------------ 4

1.5.2 Functions ------------------------------------------------ 4

2. Design overview ------------------------------------------------ 4

2.1 Design objective ------------------------------------------------ 5

2.2 Design alternative ------------------------------------------------ 5

2.3 User interface paradigms ------------------------------------------------ 5

2.4 Error detection/ Exceptional Handling ------------------------------------------------ 5

2.5 Performance ------------------------------------------------ 6

2.6 Maintenance ------------------------------------------------ 6

3. System architecture ------------------------------------------------ 6

3.1 Structure ------------------------------------------------ 7

4. Environment description ------------------------------------------------7

4.1 Time zone support ------------------------------------------------7

4.2 Language support ------------------------------------------------7

4.3 User desktop requirement ------------------------------------------------7

4.4 Server-side requirement ------------------------------------------------7

4.4.1Deployment consideration ------------------------------------------------8

4.4.2 Application server disk space ------------------------------------------------8

4.4.3 Database server disk space ------------------------------------------------8

4.4.4 Integration requirements ------------------------------------------------8

4.4.5 Network ------------------------------------------------8

4.5 Configuration ------------------------------------------------8

4.5.1 Operating system ------------------------------------------------8

5. Reference ------------------------------------------------8

**1.INTRODUCTION:-**

Bug Tracking is the process of logging and monitoring bugs or errors during software testing. It is also referred to as defect tracking or issue tracking.A bug tracking system is a software application that is designed to help quality assurance and programmers keep track of reported software bugs in their work.An efficient bug tracking system that can be mapped well to your development and quality processes is an invaluable tool .Most of the bug tracking systems support the triage of incoming bugs that is setting the priority of a bug and assigning it to a particular developer.

**1.1 Intended Audience: -**

The audience set for this project “BUG TRACKING SYSTEM” would include employees , developers and managers to keep track of reported software bugs in their work.

* 1. **Project Purpose: -**

The purpose of the Bug Tracking System is to test the application for the bugs and report it to the project manager and developer. It helps us to foresee the risk and challenges that could detail the completion a project. It is used to help business and individuals track projects.

* 1. **Key Project Objectives: -**
* Create task list with easy to use checklist.
* View and add details for each task.
* Comment on any step of a task
* Set remainders for others on specific times and dates.
* Scheduling tasks and projects which are organised by date and time.
* Adding detailed reports of the project.
  1. **Project scope and limitation: -**
* Primarily, the scope of the bug tracking system is to the perfect or unique solution to track the bugs of a solution, product or an application.The main aim of the bug tracking system to help quality assurance and programmers keep track of reported software bugs in their work.Set project timelines and milestones track project and manage teams entire workload all in one place
  1. **Functional Overview: -**

1.5.1 Following header files are included in the program:

* #include <stdio.h>
* #include <string.h>
* #include <stdlib.h>
* #include<unistd.h>
  + 1. Employee Login

Employee: Employee logins by entering userid,user role and password.If login is not successful ask user to re-enter login credentials and allow only 3 chances for trying again.

* + 1. Employee Registration:

The employee can register by entering name,email address,employeeID and phone number.

* + 1. Bug Tracking Menu:

If the employee gets successfully logged in then bug tracking menu will be displayed which contains three functions.They are

* Manage Projects
* Manage Tickets
* Manage Reports

1.5.3.1 Manage Projects:

In this function,projects created by login employee can be modified and the projects which donot have any tickets raised can be deleted.

Specifications:

* Add: Here project details will be added i.e. project\_id,description,start date,end date,number of members,department.
* Delete: The projects which do not have any tickets raised will be deleted.
* Modify: Project details can be modified.
* Quit

1.5.3.2 Manage Tickets:

In this function information about ticket is stored in “tickets.txt” file where one line corresponds to one bug.

Specifications:

* Add: Ticket details will be added.
* Close: If bugs get resolved ticket will be closed.
* Get BUG Report: Bug details will be displayed i.e. bug\_id,bug\_name and project\_id.
* Add note
* Modify: Tickets can be modified with the following operations

a)Open:

b)Assigned

c)Fixed

d)Submitted

e)Quality Checked

f)Close

* Quit

1.5.3.3 Manage Reports:

In this function report is generated as per priority of project or projects assigned by logged in employee.

Specifications:

1. Prioritize list of reports not closed
2. Prioritize list of reports assigned to logged in employee
3. Prioritize list of reports created by logged in employee.
4. Quit

# 2.Design Overview: -

Bug tracking system comprises of following modules

## Design Objectives: -

* Create task list with easy to use checklist.
* View and add details for each task.
* Comment on any step of a task
* Set remainders for others on specific times and dates.
* Scheduling tasks and projects which are organised by date and time.
* Adding detailed reports of the project.
  1. **Design Alternative****: -**

We have used linked list instead of stack & queue as Insertion and Deletions operations are fast and easier in linked list. Memory allocation is done during run-time. (i.e., no need to allocate any fixed memory.

### User Interface Paradigms: -

The Bug Tracking System gives a user an option to have its task management stored on a system file. A system always works faster than a person can. User is given an interface to create a new account , an option to deposit and check their bug in projects . A specific set of tasks are given interface to edit details of the project and send bug report to the manager

### Error Detection / Exceptional Handling: -

* If the employee doesn’t have any pre-existing account , the user has to register one else it won’t perform any functions and would give “not found” or “Invalid entry” error.
* While creating the account ,employee should first enter the name followed by phone number,emailID,employeeID,department, else it will display “Already exist” and “Invalid length ” error for the respective cases. We check the validity of the name ,email address & phone number entered with the help of exception handling .If the name entered has the length less than 5 or greater than 25 or the phone number entered is either already existing or of not length 10 digit , an error message will be flashed.

### Performance: -

### The system will work on the user’s terminal. The performance shall depend upon hardware components of the organisation and the internet connection

### Maintenance: -

Very little maintenance should be required for this setup. An initial configuration will be the only system required interaction after system is put together. The only other employee 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 result in temporary loss of data or Internet. Upgrades of hardware and software should have little effect on this project but may result in downtime.

**3.SYSTEM ARCHITECTURE: -**

**3.1 Structure Details:**

* Task Management
* View and add details for each task.
* Comment on any step of a task (or multiple steps at once).
* Set reminders for yourself or others on specific dates and times.

**4.Environment Description:**

* 1. **Time Zone Support: -** IST- Kolkata
  2. **Language Support: -** English
  3. **User Desktop Requirements: -**
* 64-bit processor, 1 GHz or faster
* At least 10 GB free hard drive space
* At least 1 GB RAM **Server**
  1. **-Side Requirements: -**
* 64-bit processor, 1 GHz or faster
* At least 2 GB free hard drive space
* At least 1GB RAM

4.4.1.Deployment Considerations: -

* + Local storage is used
  + No network latency to consider
  + To scale buy a bigger CPU, more memory, larger hard drive, or additional hardware

**4.4.2**. Application Server Disk Space: -

No such disk space is required as the program is fully functional on online

IDE(s) as well. Local Operating System is required and two txt file to store the

records of processes.

**4.4.3**. Database Server Disk Space: -

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

well. Local Operating System is required and two txt file to store the records of

processes.

**4.4.4**. Integration Requirements: -

* Language: - C
* Tools: - Valgrind, Makefile ,Strace,splint
* Complier: - gcc
* Linux Environment

**4.4.5**. Network: - End to End

**4.5 Configuration: -**

**5.5.1**. Operating System: - Linux environment

1. **Reference: -**

The references are:

* https://www.programiz.com/dsa/linked-list
* https://www.javatpoint.com/file-handling-in-c
* https://www.educative.io/answers/how-to-create-a-simple-thread-in-c