Defect Programmer Assignment

Software Requirement Specification (SRS) Document

Group 3rd

Batch: E2E-Manipal-Systems C Linux Programming PT Aug 11th Batch 2

SPRINT 1

Implementation

Project Timeline

September 28th, 2022 to October 7th, 2022

Contents

1. INTRODUCTION

The introduction of the software requirement specification provides an overview of the entire software. The entire SRS with overview description purpose, scope, tools used and basic description. The aim of this document is to gather, analyze and give an in-depth insight into the complete **Defect Programmer Assignments** application by defining the problem statement in detail. The detailed requirements of the **Defect Programmer Assignments** application are provided in this document.

1.1 Purpose: The purpose of this document is to show the requirements for the **Defect Programmer Assignments** application, in which Defects are assigned to programmers depending on the functional area of the defect and expertise of programmer.

1.2 Intended Audience: This document is intended to be read by Client.

1.3 Intended Use:
- o Development Team
- o Maintenance Team
- o Clients

1.4 Scope: This project aims to create the development of a Defect Programmer Assignments application where Clients of software development companies report defects or bugs in the software. These defects are mapped to programmers depending on their area of expertise.

2. OVERALL DESCRIPTION

The **Defect Programmer Assignment** application is used to read the defects that arise in a program and to overcome them. The details of each defect are provided in the form of text file to the application. Then, for each defect a separate thread is created. For each thread the functional area of the defect is mapped to the area of expertise of the programmer. If a programmer is found the status of defect is changed from opened to assigned. If multiple programmers are found the defect is assigned to the first one from the list. In order to avoid assigning multiple defects to a programmer before the previous defect is resolved, a mutex lock is used on the programmer once a defect is assigned to him.The solution is built as a multi-file multi-directory solution. All codes are documented well, and the coding standards are to be followed (Usability, Reliability and Availability, Performance).

2.1 Assumptions and Dependency:
- o   System should have Ubuntu Linux installed.
- o   System should have either 4GB or more RAM.

3. SYSTEM FEATURES AND REQUIREMENTS

3.1 Functionality

3.1.1 DP01 Assign_Defect: These defects are assigned to programmers depending on their area of expertise. If more than one programmer is found, the defect is assigned to the first one.

3.1.2 DP02 Read_input_files: Details about the defects are provided in text(.txt) files which need to be passed through the command line.

3.1.3 DP03 Read_and_Display_DefectsInfo: Read the details from the defect text files for each defect and display them.

3.1.4 DP04 Read_and_Display_ProgInfo: Read the details from the programmer text files for each defect and display them.

3.1.5 DP05 Defect_status_change :   All "open" defects from defects files should be processed as follows - For every defect find programmer who is expert in defect's "functional area". Assign the defect to programmer and change status of defect from "open" to "assigned". If more than one programmer is found, assign the first one.

3.1.6 DP06 Display_assigned_Defects :   Defects assigned to programmers to be displayed along with defect description, module name, functional area, filed-on date type, Emp ID and EMP Name.

3.1.7 DP07 Display_Detected_Invalid_Entry: For every invalid entry detected, this function will be called, and this will Display error messages for invalid entries and details are added to "invalidDefect.txt"

3.1.8 DP08 Display_Unassigned_Defects: It will Display all the unassigned Defects present in unassigned and details are added to "unassignedDefects.txt".

3.2 System Requirements

3.2.1. Tools to be used

i. C language

ii. POSIX Library is used for multithreading. Also, one Mutex should be used per programmer

iii. File Handling is used to read "defects" file, and "Employee" data and also to write assignments, invalid defects, and unassigned defects.

iv. Dynamic memory allocation is used for programming

v. CUnit, Valgrind, Make, and GDB are also used for making, analyzing and debugging the program

vi. System Programming

3.3 System Features:

i. **Supportability** - The system is built using C language.

ii. **Design Constraints** - The solution is built as a multi-file multi-directory solution. All codes are documented well and the coding standards are followed

iii. **Usability** – This program can be used for assigning work to Engineers as per the requirements and the domain of knowledge. The system can read any number of defect files given in the format and assign the work to the employees as per the requirement.

iv. **Reliability and Availability** – The system is available 24/7 and the user can use the system whenever he/she needs it. The user can avail of the functionalities of the program by providing the necessary files and also giving permission to read the employee data.

v. **Performance** – The system will work on the user terminal

4. DATA FLOW DIAGRAMS

Client Company — Read Defects Data → Defect Program Software
Client Company ← Change status to Valid defects — Defect Program Software
Client Company ← Invalid Defects File — Defect Program Software
Defect Program Software ← Read Employee's Data — Software Company
Defect Program Software — Unassigned Data File → Software Company

Invalid Defects
validating Defects
Valid Defects
Employee Defects
Employee's Detail
Get Defect Data
Get Valid Data
Get Employee's Data
Assigning Defects
Return Assigned Defects
Return Unassigned Defects
Saving Invalid Defects
Reading Defects
Changing Assign Status
Updating Unassignrd Defects
Employee's data
sending Employee's Detail
Defrects File
Client Company
Software Company
Invalid Defrects File
Assigned Details File
Unassigned Defects File