**Testing Management System**

Software Requirements Specification

TEAM 5

Contents

[1. Introduction 4](#_Toc379144231)

[2. Purpose 4](#_Toc379144232)

[3. Scope 4](#_Toc379144233)

[4. Definitions, Acronyms and Abbreviations 4](#_Toc379144234)

[5. References 5](#_Toc379144235)

[6. Overall Description 5](#_Toc379144236)

[6.1 Product Perspective 11](#_Toc379144240)

[6.2 Product Functions 11](#_Toc379144241)

[6.3 User Characteristics 11](#_Toc379144240)

[6.4 Constraints 11](#_Toc379144241)

[6.5 Assumptions and dependencies 11](#_Toc379144240)

[7. External Interface Requirements 7](#_Toc379144237)

[7.1 User Interfaces 11](#_Toc379144241)

[7.2 Hardware Interfaces 11](#_Toc379144240)

[7.3 Software Interfaces 11](#_Toc379144241)

[7.4 Communications Interface 11](#_Toc379144240)

[8. Functional Requirements 8](#_Toc379144238)

[8.1 Use case Diagram 11](#_Toc379144240)

[8.2 Use case Description 11](#_Toc379144241)

[9. Non Functional Requirements 11](#_Toc379144239)

[9.1 Usability 11](#_Toc379144240)

[9.2 Security 11](#_Toc379144241)

[9.3 Performance 12](#_Toc379144242)

[9.4 Capacity 12](#_Toc379144243)

[9.5 Recovery 12](#_Toc379144244)

[9.6 Availability 13](#_Toc379144245)

[9.7 Reliability 13](#_Toc379144246)

[9.8 Maintainability 13](#_Toc379144247)

[9.9 Portability 13](#_Toc379144248)

[9.10 Privacy 13](#_Toc379144249)

[10. Screen Shots 11](#_Toc379144239)

[10.1 LOGIN PAGE 11](#_Toc379144240)

[10.2 HOME PAGE 11](#_Toc379144241)

[10.3 ADD/REMOVE TESTER 12](#_Toc379144242)

[10.4 TESTCASE EXECUTION 12](#_Toc379144243)

10[.5 MODIFICATION AND DELETION OF TESTCASE 12](#_Toc379144244)

[10.6 MODIFICATION OF TESTCASE DETAILS 13](#_Toc379144245)

[10.7 REGISTER DEFECT 13](#_Toc379144246)

[10.8 SEARCH DEFECT 13](#_Toc379144247)

[10.9 VIEW DEFECT 13](#_Toc379144248)

[10.10 RECTIFY DEFECT 13](#_Toc379144249)

[10.11 REASSIGN DEFECT 13](#_Toc379144247)

[10.12 REPORT FILTERING AND EMAILING 13](#_Toc379144248)

[1013. GENERATE REPORT IN DIFFERENT FORMS 13](#_Toc379144249)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Rashmi, Anupam,  Shama, Shantan | 14/02/16 | First documentation of the SRS document for Defect and submission Management | Version 1.0 |

# Introduction

Every software developed needs a testing tool which tests the developed software for the various test cases. The objective of a testing management tool is to document all the test cases for all the possible scenarios for the software and have a consolidated report of the desired outcomes, as well as, the actual outcomes for each individual case. As a result, the testing management tool would point out the bugs in the code of the software which the programmers can fix to ensure the smooth functioning of the software without any issues or errors and the users don’t face any difficulty while using the software.

# Purpose

The purpose of this Software Requirements Specification (SRS) document is to describe the overall behavior of Testing Tool Management System (TTMS). This SRS defines and describes the operations, performance, and quality assurance requirements of the TTMS that is to be developed. This document also describes the nonfunctional requirements. It also describes the design constraints and technologies that are to be considered when the system is to be designed, and other factors necessary to provide a complete description of the requirements for the system. This Software Requirements Specification (SRS) captures the complete software requirements for the system. Requirements described in this document will be used as guidelines to develop the Testing Tool Management System

# Scope

The Testing Tool Management System will automate the following processes.

* Documenting the test cases for the given software.
* Testing of the test cases by the testing programmers and documenting the desired and actual outcomes to the test case.
* In case of difference between the desired and actual outcome, report the corresponding test case as a defect.
* Assignment of programmers to individual defects reported by the testing programmers.
* Report Generation on the test cases and the defects for measuring the effectiveness of the software developed.
* Repurpose and reuse assets across your testing cycles.
* Customize the tool for your development methodology.
* Determine test coverage and ensure tests exist for all requirements.
* Establish clear traceability between requirements, test cases, and defects.

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Designation** |
| TTMS | Testing Tool Management System |
| SRS | Software Requirements Specification |
| TCM | Test Case Management |
| SRS | Software Requirements Specification |
| PM | Project Manager |
| SRS | Software Requirements Specification |
| PID | Project Identification |
| RID | Requirement Identification |

# References

Few online Testing tools

# Overall Description

1. **Product Perspective**

Test Case Management will help manage the test cases. There will be set of test cases. Tester will enter the test case using the interface provided and system will match the actual result with expected result and store whether the output is correct or not. Testing Management system will help users to test their codes .There would be two columns one for expected output and actual output. If the actual output of user doesn’t matches with expected output then the user can submit their problem by using the form. The status of defect will show pending if the defect is not rectified and after the rectification of defect the status of problem will show problem solved. The problem of the user is registered by a unique id and is solved by skilled programmers. The solution is then posted by the programmers.

***Overview of the proposed system***

1. **Product Functions**

The following are the user requirements for the system

* + 1. There are three types of End users for the system. The first page will consist of three options for users to select. Options are

6.2.1.1 Project Manager Log In

6.2.1.2 Tester Log In

6.2.1.3 Programmer Log In

* + 1. After the user select the user type. He will be redirected to second page. Which will consist of following fields

6.2.2.1 Enter Username

6.2.2.2 Enter Password

6.2.2.3 Forgot Password

* + 1. If the user is PM. He can use the authority to assign testers or remove them for particular project. He will assign the PID and RID to the testers. Project Manager’s interface will consist

6.2.3.1 Add testers

6.2.3.2 Remove testers

6.2.3.3 Assign PID and RID to testers

6.2.3.4 Modify test case

6.2.3.5 Delete test case

6.2.3.6 Search test cases

6.2.3.7 Modify and view the test cases.

* + 1. If the user is a tester. Tester can test the software using TCM interface by entering the following details for the test case.

6.2.4.1 Project ID

6.2.4.2 Requirement ID

6.2.4.3 Test Case ID

6.2.4.4 Test Case Description

6.2.4.5 Input

6.2.4.6 Expected Output

6.2.4.7 Actual Output

* + 1. The tester assigned to the test case can enter the defect for the test case by entering the following details.
  1. Defect Description
  2. The other details given below are automatically filled.
     1. Project Name
     2. Test Case Name
     3. Tester Id –The defect details are assigned by the tester for the test case.
     4. Programmer ID –The programmer responsible for the test case.
     5. Defect open Date – Current date and time
     6. Status – Open
     7. The Programmer assigned to the test case can view the following details about the defect for the test case he has be assigned. The Project Manager can view all the defect test case details.
  3. Input
  4. Actual Output
  5. Expected output
  6. Test Case ID
  7. Project ID
  8. Requirement ID
  9. Defect ID
  10. Defect Description
  11. Tester Assigned
  12. Assigned Date and time
  13. Programmer Assigned
  14. Status of the defect
  15. Closed Date
      1. The Programmer assigned to the defect can reassign the defect to another programmer or he can delete the defect if that defect is not considered as a defect
  16. The Programmer can view the following attributes
      1. Project Name
      2. Test case id
      3. Tester id
      4. Defect id
      5. Status
  17. The other details given below can be modifies by programmer to reassign the defect.
      1. programmer name
      2. Status – open
      3. The Programmer, tester and the program manager selects the required filter for report generation. The filters are as follows:
         + 1. Date
           2. Project Name
           3. Test Case ID
           4. Assigned to
           5. Assigned by
           6. Defect ID

The Program Manager can view all the projects. The Programmer can only view the projects done by them and same condition for the testers as well.

* + 1. The Report can be generated in two data formats:
       1. PDF
       2. HTML

1. **User characteristics**

End users should have basic knowledge of computer systems to adapt to the form based user interfaces offered by the system with sufficient documentation and end user training.

* TESTER: He is the person who will test the code and finds is there any difference between actual and expected output. He can register the defect in the test cases. He can search for the defect and view the defect. And he can generate the reports.
* PROGRAMER: He is the person who rectifies the errors and gives the solution of problem. He can reassign the defect to other programmer. He can search for the defect and view the defect. And he can generate the reports.
* PROJECT MANAGER: He can assign the test case to testers and delete them. He can over right the test cases. . He can search for the defect and view the defect. And he can generate the reports.

1. **Constraints** 
   1. Migration of all existing problems registered by users to the database.
   2. Separate database for the debugged solutions issued by programmers.
   3. Solution can’t be posted without getting it tested from the tester.
   4. Interacting with requirement team and project team.
   5. Details of each test case should be stored in the database.
   6. Status should be visible to the tester, Project manager for each test case entered by the tester.
   7. PM can modify and report bug for any test case.
   8. Tester can modify, report and delete only those test cases which are assigned to him/her.
2. **Assumptions and dependencies**

1. Solution of the problems would only be issued to the users after they are successfully tested by the users.
2. Only assigned programmers would have the access to the test defects.
3. No two test cases have same IDENTIFICATION NUMBER.
4. The database will be linked with the report management database where the report of each defect will be stored.

# External Interface Requirements

1. **User Interfaces**
2. The system will provide GUI for the users.
3. Message display when we click on button.
4. Standard buttons, functions, or navigation links that will appear on every screen, such as a help button.
5. **Hardware Interfaces**
6. To use TTMS, you will need a PC (regardless of its operating system, it can be Windows, Linux) or Mac. The system requirements are
7. CPU: 2 x Intel Core 2 (2.66 Ghz, 128K cache)
8. RAM: 2GB
9. Minimum database space: 10GB
10. If system doesn’t have touch input, a keyboard will also be required by the user to provide input.
11. **Software Interfaces**
12. The system will have an interface with the problem posted by the users and solution issued by programmers.
13. **Communications Interfaces**
14. The system will use click once deployment technology of the windows .net framework where in windows form based applications can be accessed through http over the internet. The web Browsers include

* Internet Explorer 10, 11 & Edge
* Firefox (recent versions).
* Chrome, Safari, Webkit (recent versions).

# Functional Requirements

1. **Use Case Diagrams**

Tester

Project Manager

Programmer

Tester

Project Manager

Programmer

Project Manager

Tester

Programmer

1. **Use Case Scenario Description**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 1 | |
| **Use Case Name** | TCM Log In Page | |
| Summary | Tester, Programmer and PM will have their log in IDs. First they have to choose which user class they belong to. A login page will be provided. Where they can enter their user name and password for user verification. The first page consists of following   * + Tester Log In   + Program Manager Log In   + Programmer Log In   After choosing the user type they will be redirected to second page. Which consists of following   * + Enter Username text field   + Enter Password   + Forgot Password   After the user enters the details their username and password will be verified from database. In case the user forgets his password. Then on clicking the “forgot password” link the password will be sent to his registered Email ID. | |
| Preconditions | User must have his credentials in the database. | |
| Success End Condition | You have logged in Successfully Message is displayed to the user. | |
| Failed End Condition | If User could not provide his right details. | |
| Primary, Secondary Actors | Testers, Program Managers and programmers who will enter the details for verification. | |
| Trigger | - | |
| Description | Step | Action |
|  | 1 | The End user opens the first login Page. |
|  | 2 | End user will select his user type. |
|  | 3 | User will be redirected to the second page. |
|  | 4 | User will enter the user name and password for verification. |
|  | 5 | If the user name and password get verified “you have logged in successfully” message will be displayed. |
|  | 6 | In case the user forgets his username. He will click on the “forgot password” link. Password will be sent to his registered Email ID. |

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| [**Use Case name**](http://requirmentengineeringsrs.blogspot.in/) | Home Page |
| Summary | Separate login is provided for program manager, programmer and tester.  If program manager logs in then the following details are set automatically:   * + No. of test cases solved.   + No. of unsolved defects in project.   + No. of Test cases not solved in the project.   + No. of Testers   + No. of Programmers   If programmer logs in then the following details are set automatically:   * + No. of pending defects   + No. of solved defects this week in project.   + No. of projects   If the Tester logs in then the following details are set automatically:   * + No. of pending test cases.   + No. of defects raised in project.   + No. of projects   Users can also view test case, defects and generate report from the home page. After finishing their work they can successfully log out from the system. |
| Precondition | * The user must be authorized and logged in. * The Tester registers the defect if deviation arises between expected output and actual output. |
| Success End Condition | ‘***Log In Successful’*** Confirmation Message is displayed to the Tester. |
| Failed End Condition | ***‘Please fill in the required fields’*** message is displayed to the Tester when any of the given fields is blanks. |
| Primary, Secondary Actors | Tester, Programmer, Project Manager. |
| Trigger | This use case is initiated based on the registration of the defect by the Tester. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 3 | |
| **Use Case Name** | Add and remove tester | |
| Summary | Project manager can use the authority to assign testers or remove them for particular project.  By selecting Project ID, PM can assign or remove the testers for that project. | |
| Preconditions | PM will have a username and password assigned. | |
| Success End Condition | After adding or removing, Added Successfully or Removed Successfully message will be displayed. | |
| Failed End Condition | If there is some problem, and tester can’t be added then failed message will be displayed. | |
| Primary, Secondary Actors | PM to assign the testers. | |
| Trigger | This use case is initiated by the PM logging in. | |
| Description | Step | Action |
|  | 1 | The End user selects Tester or Project Manager login option |
|  | 2 | End user has to give Username and Password to access TCM system |
|  | 3 | Then user clicks on login button |
|  | 4 | The End user will then be directed to another page. |
|  | 5 | PM will select a Project ID and RID from drop down list. |
|  | 6 | Assigned testers will be displayed for that project ID and RID. |
|  | 7 | PM can select any tester to remove from that project |
|  | 8 | PM will select a tester to add tester to the particular Project ID and its module (RID). |
|  | 9 | If added or removed successfully a message will be displayed. |
|  | 10 | If failed, a message will be displayed. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 4 | |
| **Use Case Name** | Test Case Execution by Tester | |
| Summary | Tester can test the software using TCM interface by entering the following details for the test case.   * + Project ID   + Requirement ID   + Test Case ID   + Test Case Description   + Expected Output   + Actual Output   + Result | |
| Preconditions | Tester who has been authorized by the Project Manager can only login to use TCM system. | |
| Success End Condition | Test case has been Executed Successfully Message is displayed to the user | |
| Failed End Condition | Asterisk will be shown at the top-right corner of the fields to the User when any of the given fields is blank or all zeroes | |
| Primary, Secondary Actors | Testers to test the cases. | |
| Trigger | This use case is initiated by the user logging in. | |
| Description | Step | Action |
|  | 1 | The End user selects Tester or Project Manager login option |
|  | 2 | End user has to give Username and Password to access TCM system |
|  | 3 | Then user clicks on login button |
|  | 4 | The End user will then be directed to another page. |
|  | 5 | This page will consist of a table, where tester will enter details of Project ID, Requirement ID and Test Case ID in continuous columns |
|  | 6 | The Test Case Description is entered in next column |
|  | 7 | Finally, Input will be entered in next column. |
|  | 8 | If test case is successful, Executed successfully message will be displayed |
|  | 9 | If Failed message is shown, tester can click REPORT BUG button on the right end of that particular row. |
|  | 10 | Tester will be directed to the defect management page, if clicked on REPORT BUG button |
| EXTENSIONS | Step | Branching Action |
|  | A | Project Manager can change the PID and RID of testers that they are assigned to. |
|  | B | The system will retrieve the tester username, date and time of particular test case, which will be used for report generation. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 5 | |
| **Use Case Name** | Modify and deletion of test cases by tester | |
| Summary | Tester can delete and modify any test case assigned to him/her. | |
| Preconditions | Tester is already logged on the page where he can modify or delete test case. | |
| Success End Condition | He can logout or can modify any failed test case and if modified, “Modified Successfully” will be displayed. | |
| Failed End Condition | -- | |
| Primary, Secondary Actors | Tester to modify or delete the test cases. | |
| Trigger | This use case is initiated when the tester clicks on modify or delete button. | |
| Description | Step | Action |
|  | 1.1 | Tester clicks on “Modify” button |
|  | 1.2 | Tester clicks on “Delete” button |
|  | 2.1 | Can modify tabular entries |
|  | 2.2 | Can delete particular test case |
|  | 3 | Can report bug for the selected test case |
|  | 4 | If modified, data will be saved in database |
|  | 5 | If modified successfully a message will be displayed. |
|  | 6 | If failed, a message will be displayed. |
|  | 7 | Can Logout if the work is finished |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 6 | |
| **Use Case Name** | Project Manager’s authority to modify Test Cases | |
| Summary | Project manager can use the authority to view and modify any test case. | |
| Preconditions | PM is already logged on the page where he can assign testers. | |
| Success End Condition | After viewing he can logout or can modify any failed test case and if modified, “Modified Successfully” will be displayed. | |
| Failed End Condition | -- | |
| Primary, Secondary Actors | PM to modify the test cases. | |
| Trigger | This use case is initiated when the PM clicks on “view test case” button. | |
| Description | Step | Action |
|  | 1 | PM can view any test case filtered by PID and RID. |
|  | 2 | PM selects any failed test case |
|  | 3.1 | PM can report bug for that test case |
|  | 3.2 | PM can modify that test case |
|  | 4.1 | If report bug is clicked, then test case will be reported to defect management. |
|  | 4.2 | If modified, data will be saved with PM details |
|  | 5 | If modified successfully a message will be displayed. |
|  | 6 | If failed, a message will be displayed. |
|  | 7 | Can Logout if the work is finished |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 7 | |
| **[Use Case name](http://requirmentengineeringsrs.blogspot.in/)** | Register Defect | |
| Summary | The Tester accountable for the test case submits the defect in the test case by entering the ***Defect Description.***  The following details are set automatically   * + Defect ID – choose the project name combo box.   + Defect Status – OPEN   + Date and time – CURRENT DATE   + Assigned to – PROGRAMER (who designed the test case) | |
| Preconditions | * The Tester must be authorized and logged in. * The Tester registers the defect if deviation arises between expected output and actual output. | |
| Success End Condition | ‘***Defect Registration Successful’*** Confirmation Message is displayed to the Tester. | |
| Failed End Condition | ***‘Please fill in the required fields’*** message is displayed to the Tester when any of the given fields is blanks. | |
| Primary, Secondary Actors | Tester, Programmer, Project Manager. | |
| Trigger | This use case is initiated based on the registration of the defect by the Tester. | |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1 | The Tester clicks on the Defect Registration button. |
|  | 2 | The Defect id is auto generated by the system and displayed to the Tester. |
|  | 3 | The Defect status is set as ***OPEN*** and displayed to the tester. |
|  | 4 | The Programmer assigned to the defect is also shown. |
|  | 5 | The Tester enters the Defect Description in the specified text box. |
|  | 6 | The Tester Clicks on the Submit Button. |
|  | 7 | The Defect open Date is set as the current date and time. |
| **EXTENSIONS** | **Step** | **Branching Action** |
|  | 1a  1b  1c    1d | The system updates the data into the database.  The system retrieves the Tester id and updates it with the Defect Registration details into the database.  The defect is assigned automatically to the programmer in the project using the project database.  Email is sent to the programmer who is assigned the defect. |
|  | 2a | Escalation Procedure is followed and mail is sent to Project Manager the along with the tester in cc about the status of the defect if the status remains as open for 24hrs. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 8 | |
| [**Use Case name**](http://requirmentengineeringsrs.blogspot.in/) | Search Defect | |
| Summary | **TESTER:**  The tester can search using various filters :  1.date  2.defect id  3.defect\_name  4.assigned programmer  5.rid  \*The tester can only see the defect any programmer works on.  **PROJECT MANAGER:**  The Program manager can search using various filters :  1.date  2.defect id  3.defect\_name  4.assigned programmer  5.assigned Tester  6.rid  \*The project manager can see all testcases under his project  .**PROGRAMMER:**  The programmer can search using various filters :  1.date  2.defect id  3.defect\_name  4.assigned Tester  5.rid  \*The programmer can only see the defect any tester works on. | |
| Preconditions | * The Programmer,tester and project manager must be authorized and logged in. | |
| Success End Condition | All the defects are displayed on the screen which are filtered according to selected parameter. | |
| Failed End Condition | ***No defects found if the defect of particular name is not there.*** | |
| Primary, Secondary Actors | Programmer, Project Manager.,tester | |
| Trigger | This use case is initiated based on the submitting the solution for the defect by the Programmer. | |
|  |  | |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1.1 | The user selects the particular parameter from the combo box |
|  | 1.2 | The defects get displayed on the screen filtered by the parameter selected.. |
|  | 1.3 | The user again selects the particular defect |
|  | 4 | The Tester Clicks on the Submit Button. |
|  | 4 | The Tester Clicks on the Submit Button. |
| **EXTENSIONS** | **Step** | **Branching Action** |
|  | 1 | The system updates the data into the database. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 9 | |
| [**Use Case name**](http://requirmentengineeringsrs.blogspot.in/) | **View Defect** | |
| Summary | * When new defect is assigned, the status of the defect should either be ***open*** or ***in progress***. The User can view the following details of the defect.   + Input   + Actual Output   + Expected output   + Assigned Date and time   + Test Case ID   + Project ID   + Requirement ID   + Defect ID   + Defect Description   + Tester Assigned   + Defect Solution   + Closed Date | |
| Preconditions | * The Programmer must be authorized and logged in. * The Tester must be authorized and logged in and can only view the defects that have been submitted by him. * The Project Manager must be authorized and logged in and can view all the defects submitted of all the conditions. * The list of defects will be shown in the page and the User must click on the ***view*** button of the particular defect from the list. | |
| Success End Condition | The Details of the defect will be displayed to the User. | |
| Failed End Condition | Connection to the Database Failed – Please contact System Maintenance For assistance Message will be displayed. | |
| Primary, Secondary Actors | Programmer, Project Manager, Tester. | |
| Trigger | This use case is initiated based on the clicking the view button for the particular defect by the User. | |
| **DESCRIPTION** | **Step** | **Action (Programmer)** |
|  | 1 | The list of all defects along with the defect status is displayed to the Programmer for a maximum of ten crimes per page. |
|  | 2 | The Programmer clicks on the Defect solution ***view*** button for a particular defect. |
|  | 3 | The Programmer changes the defect status to ***in-progress*** if it is still open. |
|  | 4 | The Programmer clicks the back button after viewing the details. |
| **DESCRIPTION** | **Step** | **Action (Tester)** |
|  | 1 | The list of all defects along with the defect status which the tester has submitted is displayed to the Tester for a maximum of ten crimes per page. |
|  | 2 | The Tester clicks on the Defect solution ***view*** button for a particular defect. |
|  | 4 | The Tester clicks the back button after viewing the details. |
| **DESCRIPTION** | **Step** | **Action (Project Manager)** |
|  | 1 | The list of all defects along with the defect status is displayed to the Project Manager for a maximum of ten crimes per page. |
|  | 2 | The Project Manager clicks on the Defect solution ***view*** button for a particular defect. |
|  | 3 | The Project Manager changes the defect status to ***in-progress*** if it is still open. |
|  | 4 | The Project Manager clicks the back button after viewing the details. |
| **EXTENSIONS** | **Step** | **Branching Action** |
|  | 1 | The system updates the defect status into the database. |
|  | 2 | For Tester: Escalation Procedure is followed and mail is sent to the Project manager along with the Programmer in cc about the status of the defect if the status remains as ***in progress*** for more than 2days. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 10 | |
| [**Use Case name**](http://requirmentengineeringsrs.blogspot.in/) | Defect Rectification | |
| Summary | * The Programmer assigned to the defect submits the solution for the defect by entering the following details   + Solution code for the test case   + Date and time   + Defect status   + Upload File | |
| Preconditions | * The Programmer must be authorized and logged in. * The Programmer submits the solution to the test case after solving the defect that is assigned to him. | |
| Success End Condition | ‘***Defect Solution submitted Successfully’*** Confirmation Message is displayed to the Programmer. | |
| Failed End Condition | ***‘Please fill in the required fields’*** message is displayed to the Programmer when any of the given fields is blanks. | |
| Primary, Secondary Actors | Programmer, Project Manager. | |
| Trigger | This use case is initiated based on the submitting the solution for the defect by the Programmer. | |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1 | The Programmer clicks on the Defect solution ***submit*** button. |
|  | 2 | The Defect status is set as ***CLOSED*** and displayed to the programmer. |
|  | 3 | The programmer enters the Defect Solution Description in the specified text box and uploads a file if required. |
|  | 4 | The Tester Clicks on the Submit Button. |
|  | 5 | The Defect close Date is set as the current date and time. |
| **EXTENSIONS** | **Step** | **Branching Action** |
|  | 1a  1b  1c | The system updates the data into the database.  The system retrieves the Programmer id and updates it with the Defect submission details into the database.  Email is sent to the project Manager with cc to all programmers and testers about completion of the defect. |
|  | 2 | Escalation Procedure is followed and mail is sent to the Project manager along with the Programmer in cc about the status of the defect if the status remains as ***in progress*** for more than 2days. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 11 | |
| [**Use Case name**](http://requirmentengineeringsrs.blogspot.in/) | **Re-assign Defect** | |
| Summary | * The Programmer can view the following attributes: * Tester id: * Project Name * Test Case id: * Defect id: * The programmer can modify   programmer name   * Programmer can submit defect and can delete the entire defect. * When programmer submit defect status will open. * Open date is now current date. | |
| Preconditions | * The Programmer must be authorized and logged in. * Click on view defect. Assign tester. | |
| Success End Condition | ‘***Defect Modified Successfully’*** Confirmation Message is displayed to the Programmer. | |
| Failed End Condition | ***‘Please check defect again’*** message is displayed to the Programmer when any of the given fields is wrong. | |
| Primary, Secondary Actors | Programmer, tester. | |
| Trigger | This use case is initiated based on submitting the modification of defect by the Programmer. | |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1 | Programmer click on Re-assign to view the defect. |
|  | 2 | Select any of programmers from drop down box. |
|  | 3 | Programmer changes its status to Open from In progress. |
|  |  | The Defect open Date is set as the current date and time.  Programmer clicks on re-assign, it re-assigns the defect. |
| **EXTENSIONS** | **Step** | **Branching Action** |
|  | 1a  1b | The system updates the data into the database.  Email is sent to the project Manager with cc to all programmers and testers about modification of the defect. |
|  | 2 | Escalation Procedure is followed and mail is sent to the Project manager along with the Programmer in cc about the status of the defect if the status remains as ***in progress*** for more than 2days. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 12 | |
| [**Use Case name**](http://requirmentengineeringsrs.blogspot.in/) | Filter and Emailing Function | |
| Summary | * + The user, program manager uses the filters to narrow down the database results   + The filters change according to the key selected in filter so that further refine options may come.   + An email generation straight to the tester can be sent | |
| Preconditions | * The Programmer must be authorized and logged in | |
| Success End Condition | The program manager, tester is able to view the information of all the test cases using the desired filters and is able to contact them. | |
| Failed End Condition | Selection of filter is a must. | |
| Primary, Secondary Actors | Programmer, Project Manager, Tester | |
| Trigger | This use case is initiated based on selecting the desired filter by the programmer, tester or the Program Manager. | |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1 | The tester, programmer or the program manager enters the username and password details. |
|  | 2 | The desired filter is selected. Then select Go. |
|  | 3 | The desired project names will be displayed with Tester ID and status |
|  | 4 | Any message can be sent to the tester via mail. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | 13 | |
| [**Use Case name**](http://requirmentengineeringsrs.blogspot.in/) | Generating report in different formats | |
| Summary | * The file for report generation can be created in several formats like HTML and PDF. * This is done using PHP Report Generator. * PHP Report Generator is a PHP class that generates customizable HTML reports with data from the result set of any MySQL SELECT query. * Just pass the query and a MySQL database connection resource id, customize several look and feel aspects of your report like text color, header color, table border, size, etc. Then call generate Report function. | |
| Preconditions | * The Programmer must be authorized and logged in and he/she should select the required filters. | |
| Success End Condition | The report is generated in the required format.. | |
| Failed End Condition | Query not generated properly from the database using the variables. | |
| Primary, Secondary Actors | Programmer, Project Manager, Tester | |
| Trigger | The report is generated by selecting the required format. | |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1 | After the tester, programmer or the program manager enters the username and password details and selects the desired filter, go to Contact Tester. Or select continue. |
|  | 2 | The page will be displayed for generating the report in PDF or HTML format. |
|  | 3 | Select the required format and softcopy is generated |
|  | 4 | The user can then either Logout or Go back. |

# Non Functional Requirements

This section describes in detail all the non-functional requirements

## Usability

* 1. The system shall allow the users to access the system from the Internet.
  2. online help will be available for the system
  3. The end users will be able to able to adapt to the system with a minimum training of 40 hours
  4. Key board short cuts will be available for all functions of the system

## Security

1. Login requirements -
   1. The testers and project managers will be provided access to the system after they are registered into the database.
   2. While logging in the system for the first time, the testers and project managers will be provided an ID and a password.
   3. On logging in, they can set a new password
2. Password requirements
   1. Password will be case-sensitive.
   2. Password must have at least 6 characters.
3. Inactivity timeouts
4. System should timeout when there is no activity for 10 minutes.

## Performance

1. Response time
2. The response time will be less than 10 seconds for almost all the processes performed in the system.

## Capacity

1. Storage
2. Hard disk space –

100 GB – Content

50 GB – Transaction Logs

## Recovery

1. Recovery time scales
2. The system will be recovered within 12 hours from the down time
3. Backup Frequencies
4. Details of all the processes carried out by the testers, project manager will be stored in the back-up tapes.
5. All the back-up tapes will be provided to the Project Manager.
6. The back-up data will be updated every 10 days.

## Availability

1. Hours of operation
2. The system will be available on all days 24\*7

## Reliability

1. Mean Time between Failures
2. The mean time between failures for the system will be 90days

## Maintainability

1. Mean Time to Recovery
2. The Mean Time To Recovery (MTTR) shall not exceed one day.

## Portability

1. OS requirements

The system will run on windows XP / Vista / 7 / 8 / 8.1 / 10 and on MAC OS and on LINUX.

1. Browser requirements

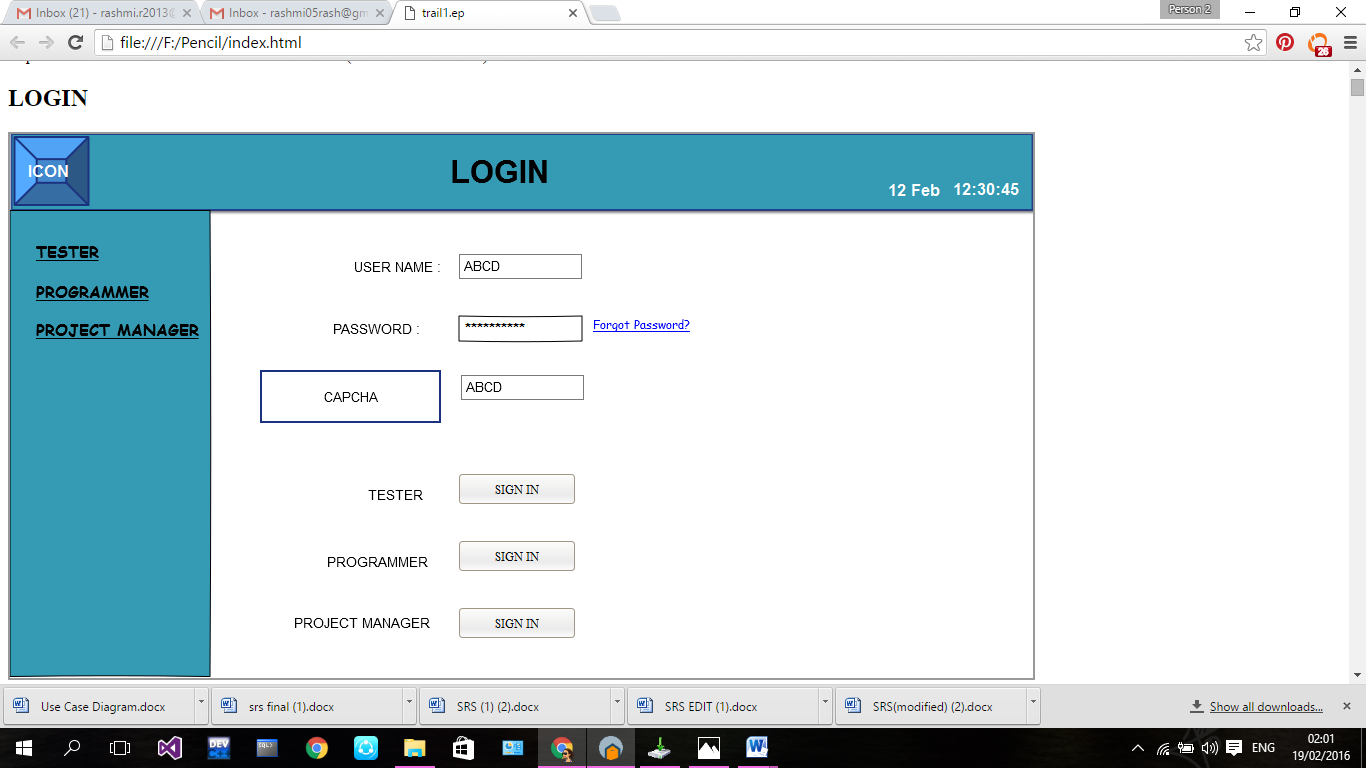
The system will run on Internet Explorer, Internet Edge, Mozilla Firefox, Google Chrome, Safari and UC browser.

## Privacy

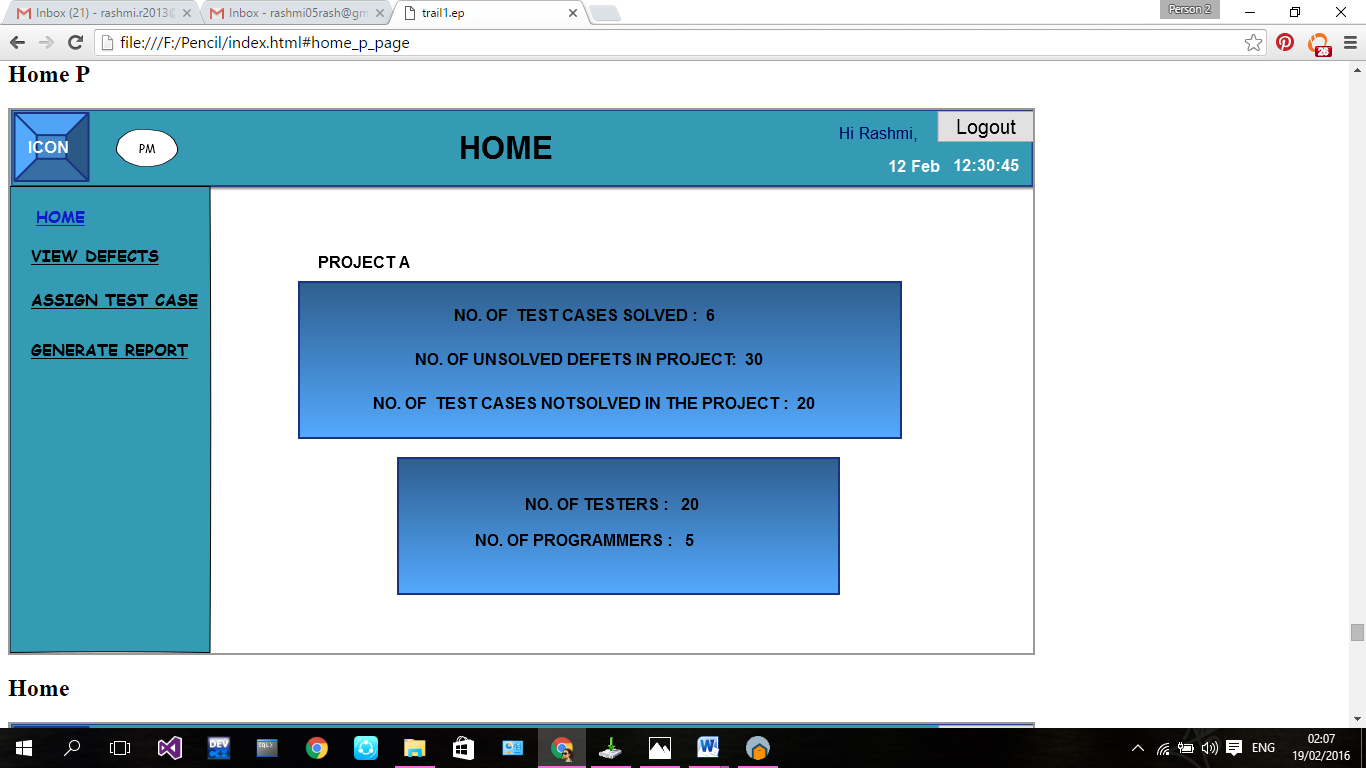
1. Reports cannot be accessed by end users who are not given access to the reports
2. No two users would be able to see their problems.
3. Every problem registered by would have unique id.

# Screen Shots

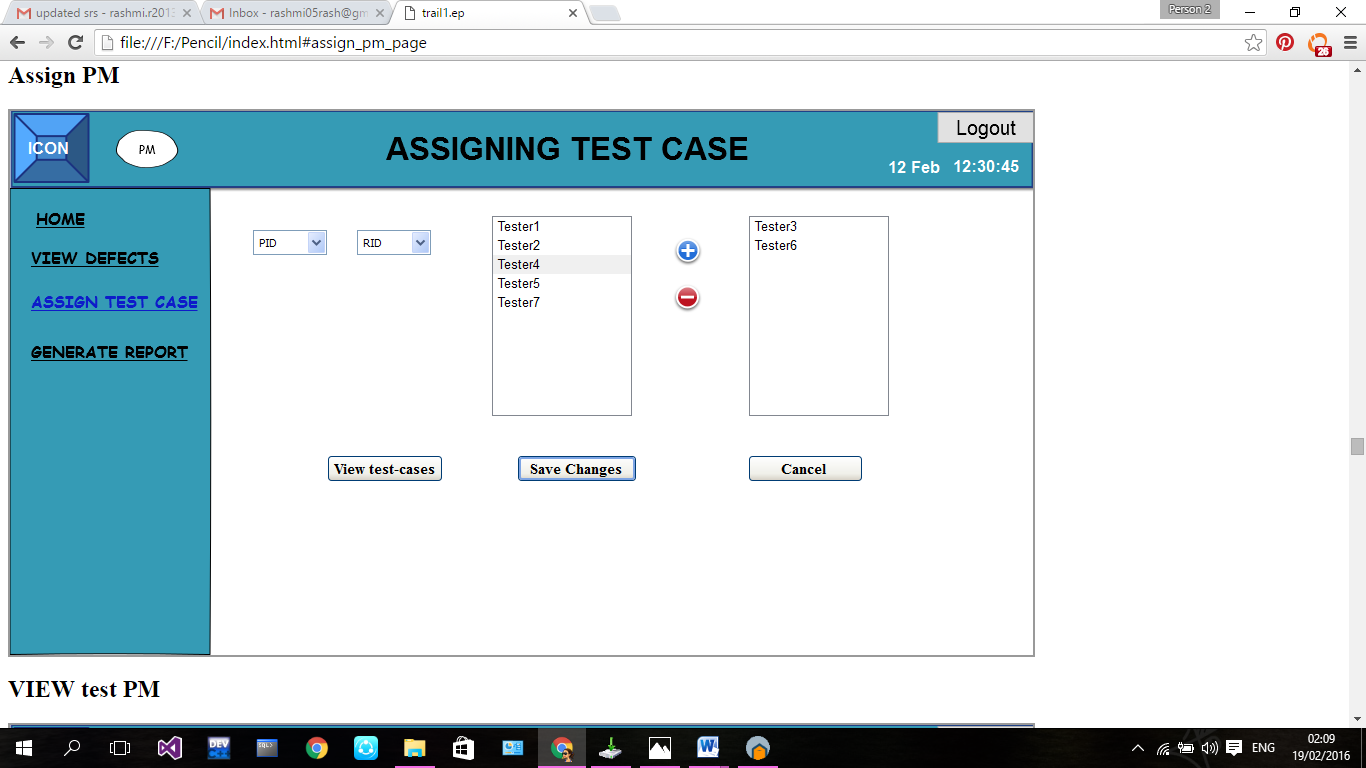
## LOGIN PAGE - SUMIT



## HOME PAGE - SATHYAM

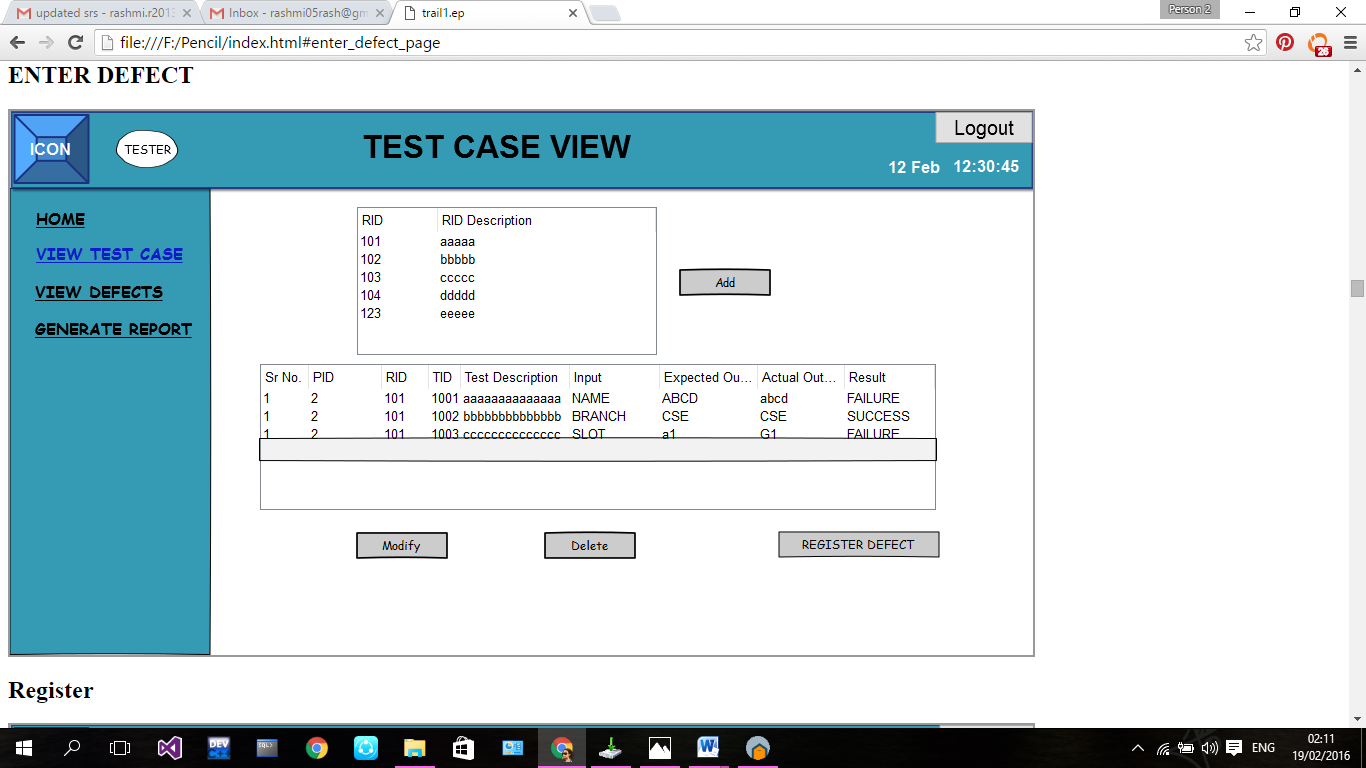


## ADD OR REMOVE TESTER - SWAROOP

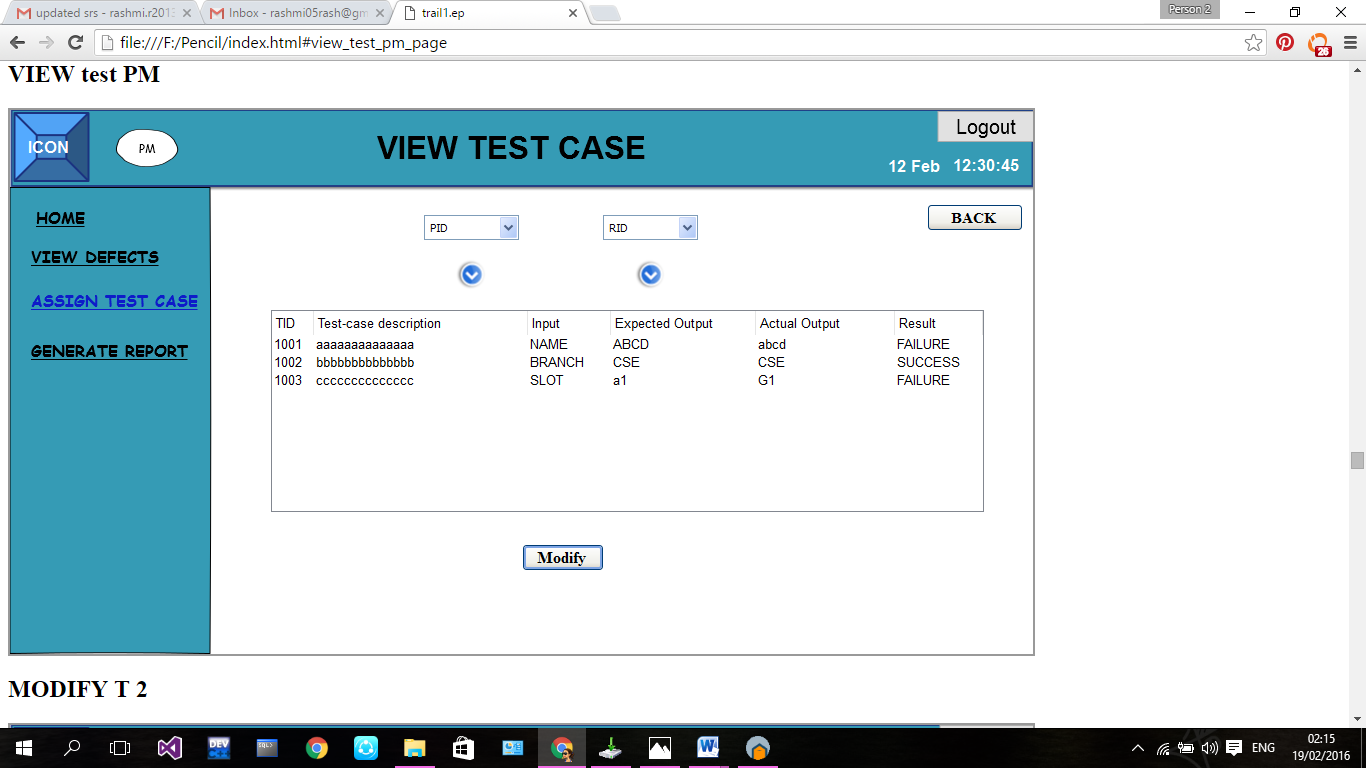


## TEST CASE EXECUTION - RAJEEV–

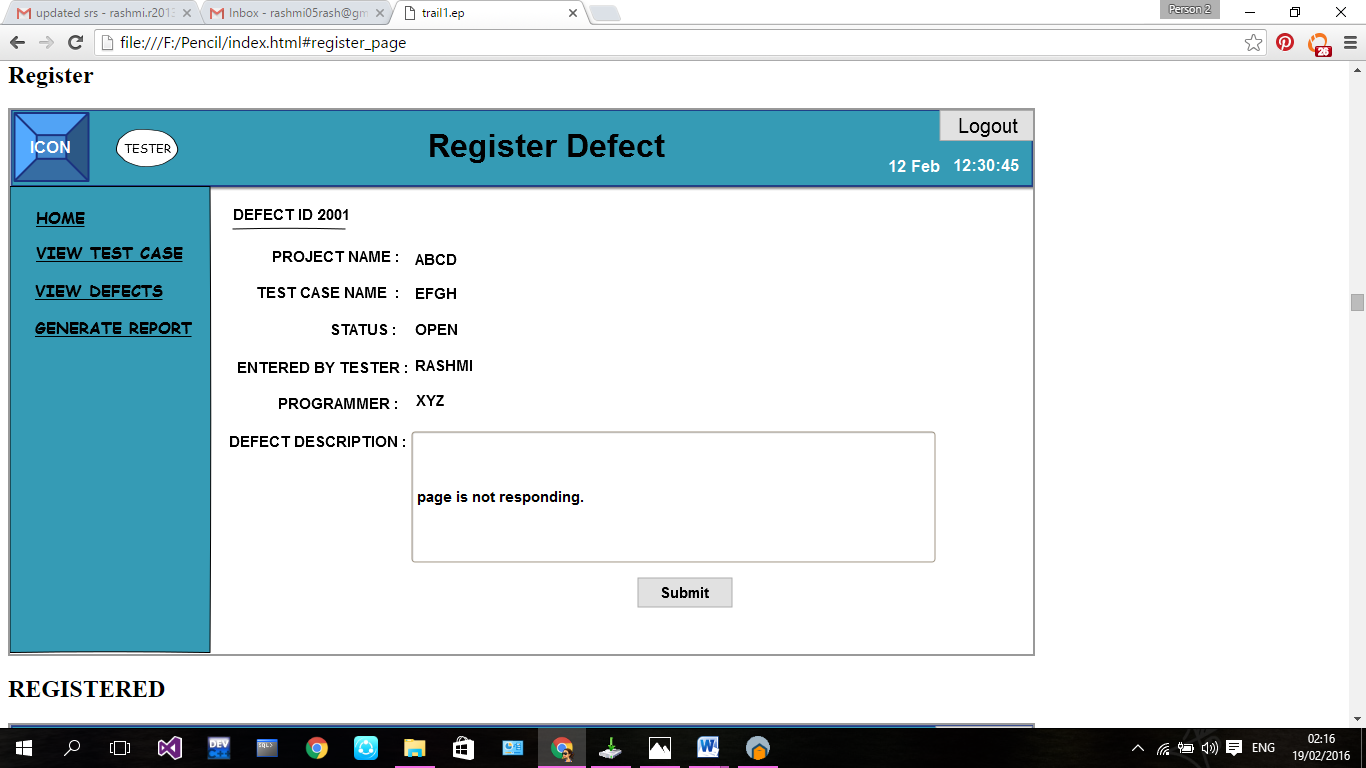
## MODIFICATION AND DELETION OF TEST CASE- RAJEEV



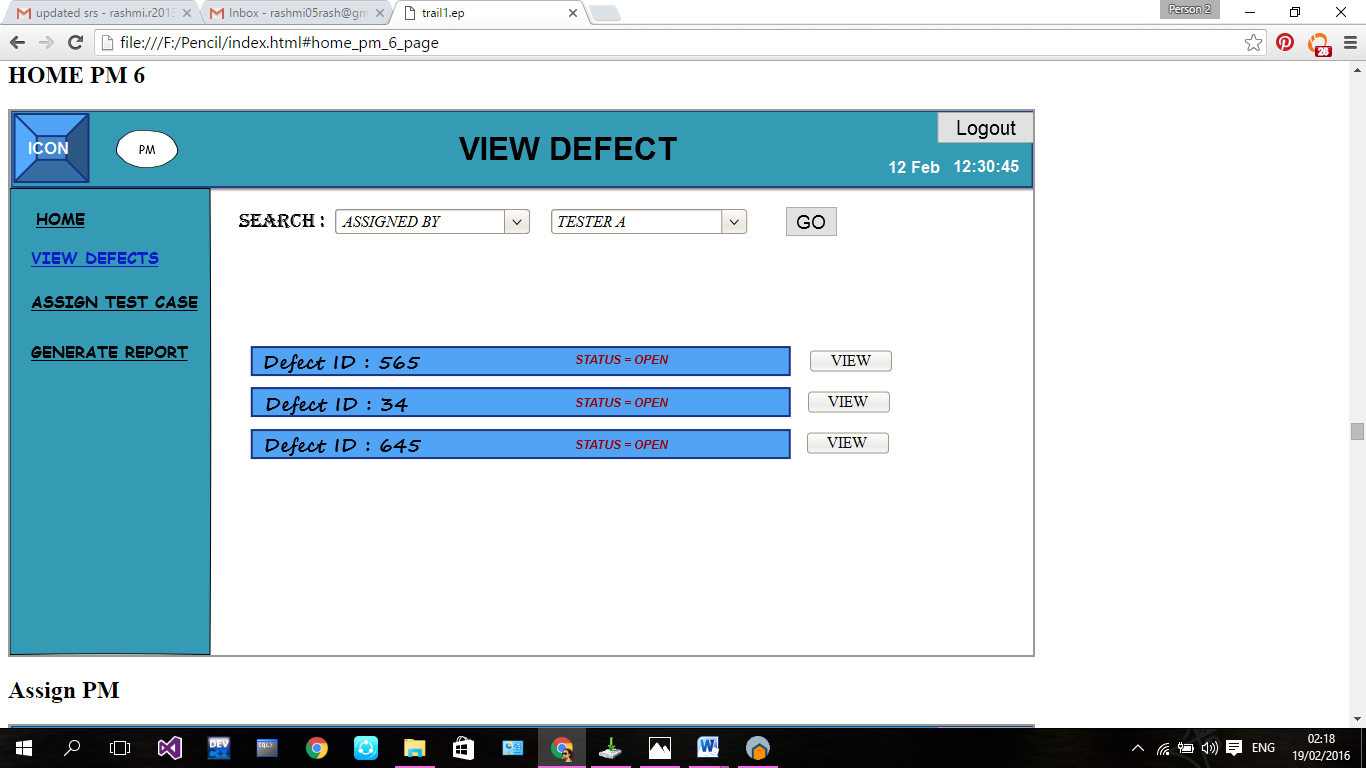
## MODIFY TESTCASE AND OVERWRITE- AKHIL



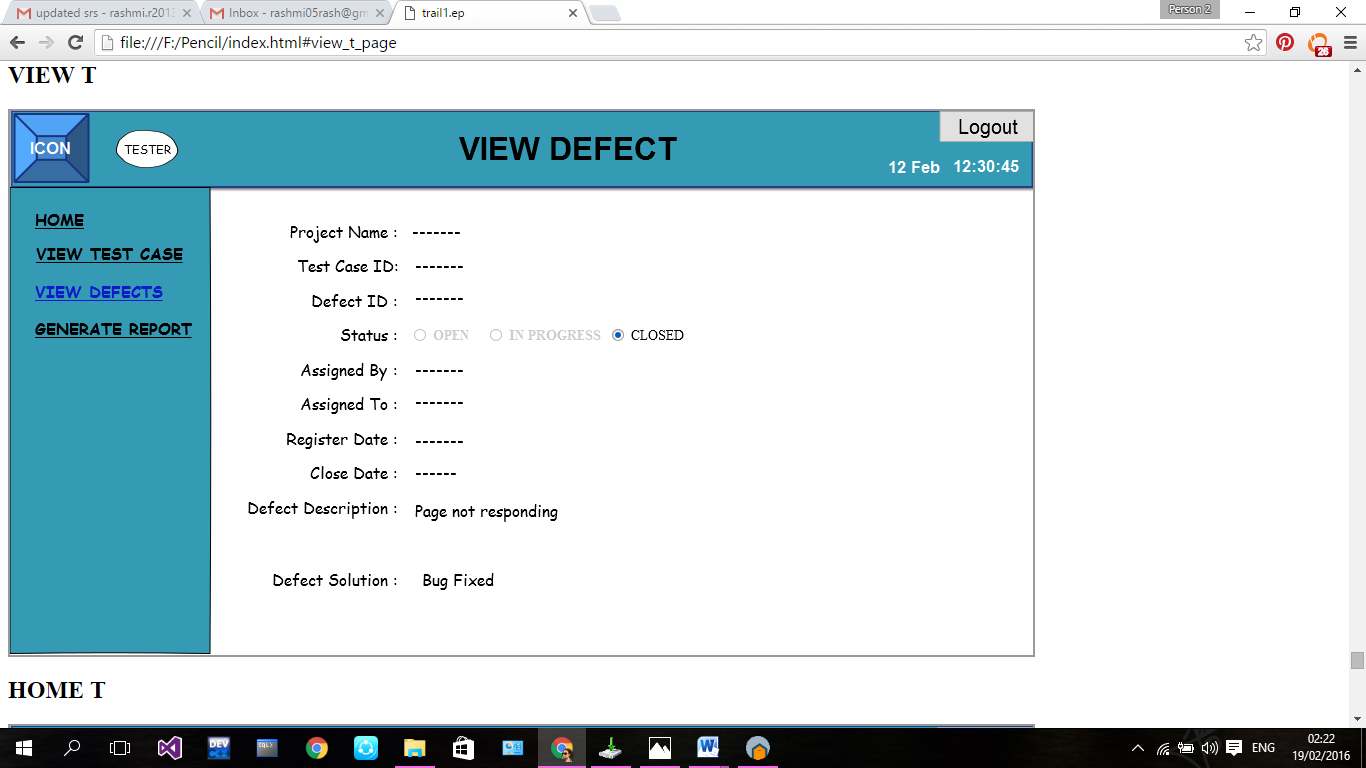
## REGISTER DEFECT- RASHMI



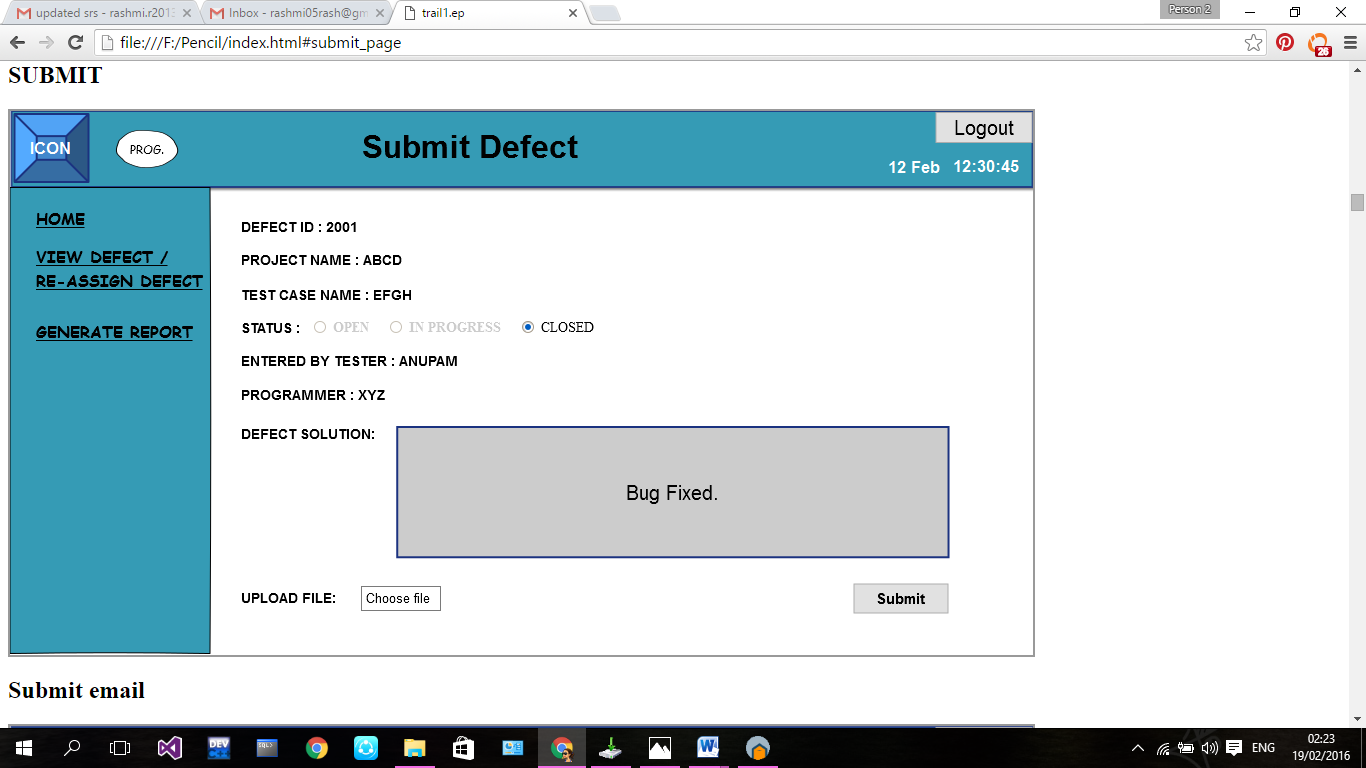
## SEARCH DEFECT- ANUPAM



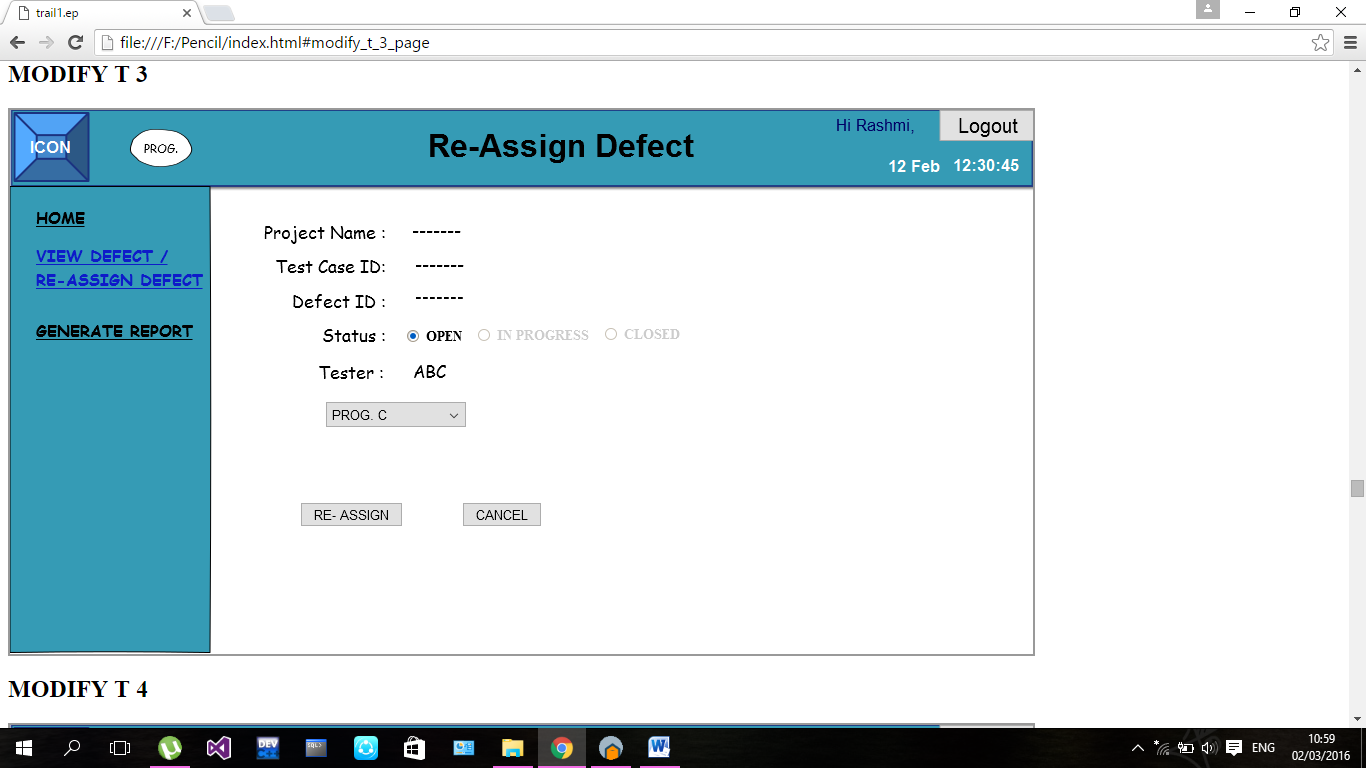
## VIEW DEFECT- SHANTAN



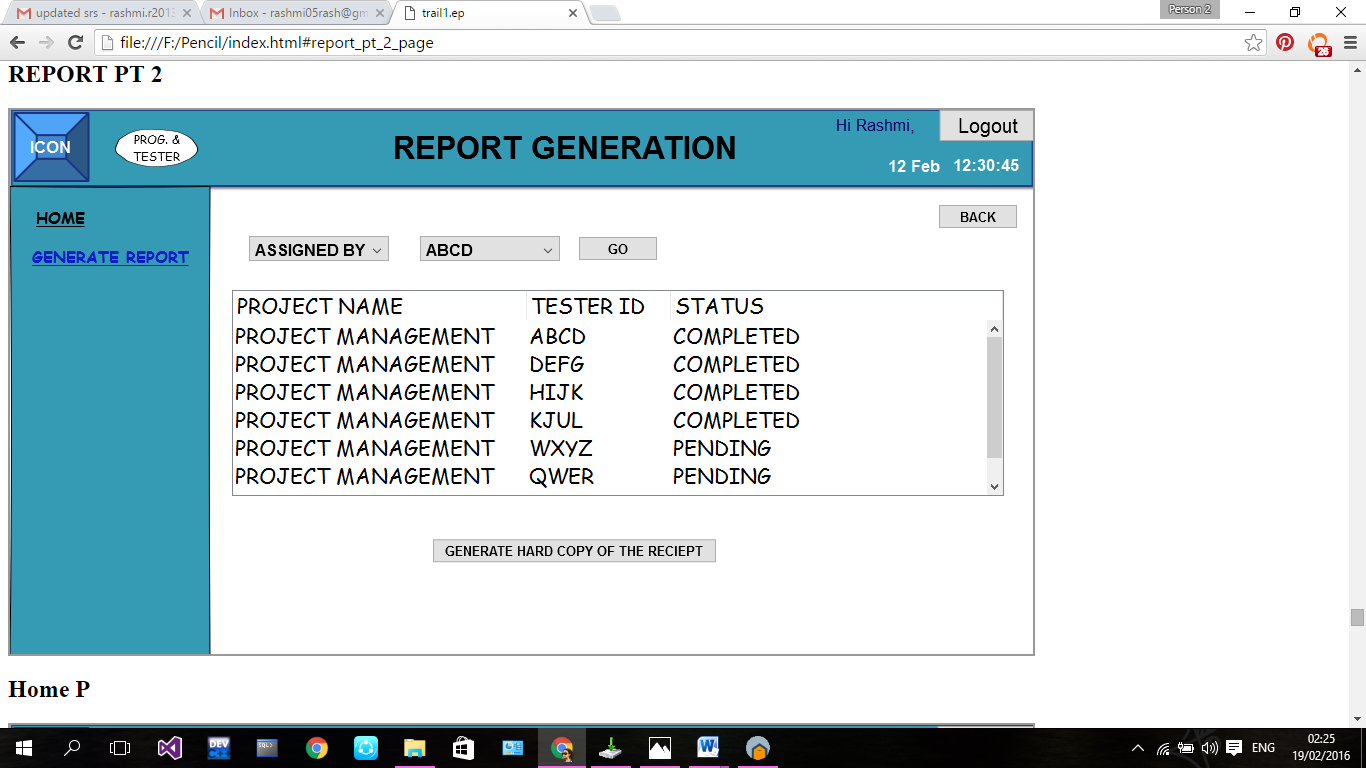
## SUBMIT SOLUTION TO DEFECT-RASHMI



## REASSIGN DEFECT-SHAMA



## REPORT FILTERING AND EMAILING - ABHINAV



## GENERATION OF REPORTS IN DIFFERENT FORMS-PRABJIT

## 