Helwan university

Faculty of engineering

* Prepared by:
* Mahmoud Ehab Mahmoud Abdeen.
* Ahmed Khaled Mohana Tohamy.
* Mariam Mohamed Abdelmonem Ismail.

Under supervision of

DR/Manal showman

Table of Contents:

[Team members: 4](#_Toc112002441)

[List of Figures: 4](#_Toc112002442)

[1. Preface: 5](#_Toc112002443)

[1.1 Document Purpose: 5](#_Toc112002444)

[1.2 Target Users: 5](#_Toc112002445)

[1.3 Revision History: 5](#_Toc112002446)

[2. Introduction: 5](#_Toc112002447)

[2.1 Purpose: 5](#_Toc112002448)

[2.2 Scope: 6](#_Toc112002449)

[2.3 Overview: 6](#_Toc112002450)

[3. Glossary: 7](#_Toc112002451)

[3.1 Acronyms, definitions, and abbreviations: 7](#_Toc112002452)

[4. System Users: 7](#_Toc112002453)

[4.1 System stakeholders: 7](#_Toc112002454)

[4.2 Users objectives: 8](#_Toc112002455)

[5. User Requirements definitions: 8](#_Toc112002456)

[5.1 System Functions: 8](#_Toc112002457)

[5.2 Constraints: 9](#_Toc112002458)

[6. System Architecture: 9](#_Toc112002459)

[7. System Functional requirements: 10](#_Toc112002460)

[7.1 Forgot password: 10](#_Toc112002461)

[7.2 View users activity table: 10](#_Toc112002462)

[7.3 View login attempts table: 10](#_Toc112002463)

[7.4 View users table: 10](#_Toc112002464)

[7.5 Add new user: 10](#_Toc112002465)

[7.6 Update existing user information: 10](#_Toc112002466)

[7.7 Create project & assign project to users: 10](#_Toc112002467)

[7.8 View project table & analyze project progress: 10](#_Toc112002468)

[7.9 View bug table: 11](#_Toc112002469)

[7.10 Add bug details (types, severity and status): 11](#_Toc112002470)

[7.11 Reset password: 11](#_Toc112002471)

[7.12 View assigned projects table & view the other assigned developers in the same projects: 11](#_Toc112002472)

[7.13 View assigned bugs table: 11](#_Toc112002473)

[7.14 Update bug status: 11](#_Toc112002474)

[7.15 Report bug: 11](#_Toc112002475)

[7.16 Update bug details: 11](#_Toc112002476)

[7.17 Delete bug: 11](#_Toc112002477)

[7.18 View an overview of bugs status in a specific project: 11](#_Toc112002478)

[8. Interface requirements: 12](#_Toc112002479)

[8.1 User interfaces: 12](#_Toc112002480)

[8.2 Software Interfaces: 20](#_Toc112002481)

[9. Non-functional requirements: 20](#_Toc112002482)

[9.1 Availability: 20](#_Toc112002483)

[9.2 Security: 20](#_Toc112002484)

[9.3 Maintainability: 20](#_Toc112002485)

[9.4 Portability: 20](#_Toc112002486)

[9.5 Usability: 21](#_Toc112002487)

[9.6 Efficiency 21](#_Toc112002488)

[10. System Models and Diagrams: 21](#_Toc112002489)

[11. System Evolution: 21](#_Toc112002490)

[12. Future Work: 21](#_Toc112002491)

[13. Time Plan: 22](#_Toc112002492)

[13.1 Work Breakdown Structure: 22](#_Toc112002493)

[13.2 Gantt chart: 22](#_Toc112002494)

[14. Appendices: 22](#_Toc112002495)

[15. References: 22](#_Toc112002496)

# Team members:

|  |  |  |
| --- | --- | --- |
| Name | Code | Work in SRS |
| Ahmed Khaled Mohana Tohamy. | 20026 | * Introduction. * System architecture. * System non-functional requirements. |
| Mahmoud Ehab Mahmoud Abdeen. | 20010 | * Preface. * User requirements definitions. * Interface requirements. |
| Mariam Mohamed Abdelmonem Ismail | 20047 | * System users. * System functional requirements. * System evolution. * Future work. * Work Plan. |

# [List of Figures:](file:///C:\Users\moham\Downloads\SRS%20example%20(1).docx#_bookmark0)

[Figure 1: System Modules hierarchy 6](#_Toc112011969)

[Figure 2: System Functions hierarchy 6](#_Toc112011970)

[Figure 3: System Architecture 10](#_Toc112011971)

[Figure 4:Intro page 12](file:///G:\material%20kolya\level%203\summer\software%20engineering\project\my%20files\documents\SRS.docx#_Toc112011972)

[Figure 5:Login Page 12](file:///G:\material%20kolya\level%203\summer\software%20engineering\project\my%20files\documents\SRS.docx#_Toc112011973)

[Figure 6:Admin page: Activity &login attempts table 13](#_Toc112011974)

[Figure 7:Admin page: Users table 13](#_Toc112011975)

[Figure 8:Admin Page: Projects table 14](file:///G:\material%20kolya\level%203\summer\software%20engineering\project\my%20files\documents\SRS.docx#_Toc112011976)

[Figure 9:Admin Page: User information 14](file:///G:\material%20kolya\level%203\summer\software%20engineering\project\my%20files\documents\SRS.docx#_Toc112011977)

[Figure 10: Admin Page: Bugs Table 15](#_Toc112011978)

[Figure 11:Developer Page: Dashboard 15](#_Toc112011979)

[Figure 12:Developer Page: Projects Table 16](#_Toc112011980)

[Figure 13:Developer Page: Bugs Table 16](#_Toc112011981)

[Figure 14:Tester Page: Dashboard 17](#_Toc112011982)

[Figure 15:Tester Page: Projects Table 17](#_Toc112011983)

[Figure 16:Tester Page: Assigned people to the same project 18](file:///G:\material%20kolya\level%203\summer\software%20engineering\project\my%20files\documents\SRS.docx#_Toc112011984)

[Figure 17:Tester Page: Bugs Panel 18](#_Toc112011985)

[Figure 18:Tester Page: View all the bugs in a specific project 19](file:///G:\material%20kolya\level%203\summer\software%20engineering\project\my%20files\documents\SRS.docx#_Toc112011986)

[Figure 19:Tester Page: View Bug Details 19](file:///G:\material%20kolya\level%203\summer\software%20engineering\project\my%20files\documents\SRS.docx#_Toc112011987)

[Figure 20:Tester Page: View Overview of Bugs status 20](#_Toc112011988)

[Figure 21:context diagram 21](#_Toc112011989)

# Preface:

## Document Purpose:

* Through this document, a complete and detailed specification of the Bug Tracker System will be provided, making it possible to understand the ins and outs of the software and what each user will be able to achieve by using it.
* In the first section, an overview of the system will be elaborated upon, and then each part of the system will be discussed in more detail in the following sections.

## Target Users:

* This document is authored by system engineers based on the requirement obtained from the senior software engineers, developers and testers of the company that issued the development of the system.
* The document is to be approved and acknowledged by the CEO, senior SW engineers, developer, testers, and the crew working on developing the project.

## Revision History:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Description** | **Date** |
| **0.1** | All members of the team | Initial | 23-7-2022 |
| **0.2** | Mariam Mohamed | Some functional requirements changes | 3-8-2022 |
| **0.3** | Mahmoud Ehab | Some changes in system functions | 6-8-2022 |
| **0.4** | Ahmed Khaled | Some changes in non-functional requirements | 9-8-2022 |
| **0.5** | Mariam Mohamed | Work plan | 12-8-2022 |
| **0.6** | Mahmoud Ehab | Minor changes in system functional requirements | 21-8-2022 |

# Introduction:

## Purpose:

* purpose of the software is to track bugs and allow the end user to manage bugs and store them if they ever appear in other projects.

## Scope:

Diagram

Description automatically generated

Figure : System Modules hierarchy

Diagram

Description automatically generated

Figure : System Functions hierarchy

## Overview:

* The document is organized as follows:
* an overview description of the bug tracker system and its high-level functions are presented (section 2.1 and 2.2).
* section 4 state types of users who can use bug tracker.
* Section 5 in the document provides user requirement definitions.
* Section 6 in the document provides system architecture.
* Section 7 in the document provides a detailed description of the system functions and requirements.
* Section 8 in the document provides interface requirement.
* Section 9 in the document provides the system non functions and requirements.
* Section 10 presents some helping information and diagrams that will facilitate the understanding of the contents.
* Section 11 in the document provides system evolution.
* Section 12 in the document provides future work.
* Section 13 in the document provides time plan.
* Section 14 in the document provides appendices.
* Finally, Section 15 in the document provides references.

# Glossary:

## Acronyms, definitions, and abbreviations:

* **FOEHU**: Faculty of Engineering, Helwan University.
* **BTS:** Bug Tracker System.
* **User:** Developer or tester who uses the application.

# System Users:

## System stakeholders:

* System Engineer.
* Responsible for gathering requirements.
* Responsible for development.
* Responsible for support.
* Administrator:
* Adds new users.
* Updates users’ information.
* Deletes users..
* Creates projects.
* Assigns projects to users.
* Deletes projects.
* Adds bug types and severity.
* Monitors user activity.
* Views failed login attempts.
* Developer:
* Views the list of assigned projects.
* Responsible for updating bug status.
* Resets and edits his information like passwords.
* Tester:
* View the list of assigned projects.
* Adds bug details.
* View bug status.

## Users objectives:

* System Engineer.
* Gains experience in software engineering and developments.
* Administrator:
* Follows up on the work of the users.
* Developer:
* Simplification in finding the assigned work.
* Prevent assigned work mixing.
* Work in an organised environment.
* Tester:
* Simplification in finding the assigned work.
* Prevent assigned work mixing.
* Work in an organised environment.

# User Requirements definitions:

## System Functions:

* **Login:**
  + 1. Forgot password.
* **Administrator:**
  + 1. View user activities table.
    2. View login attempts table.
    3. View users table.
    4. Add new user.
    5. Update existing user information.
    6. Delete users.
    7. View projects table.
    8. Create projects & Assign project to user.
    9. View bugs table.
    10. Add bug details (types and severity).
* **Developer:**
  + 1. Reset password.
    2. View assigned projects table & view the other assigned developers in the same projects.
    3. View assigned bugs table.
    4. Update bug status.
* **Tester:**
  + 1. View assigned projects table.
    2. View users of each project assigned to him.
    3. View assigned bugs table.
    4. Report bug.
    5. Update bug details.
    6. Update bug status.
    7. Delete bug.
    8. View an overview of bugs status in specific project.

## Constraints:

1. Company Policies:
   * Users must be limited by permissions to insure data confidentiality.
2. Cultural Constraints:
   * Code must follow company standards.
3. Technological Limitations:
   * Integration with Git is required to allow database synchronization.

# System Architecture:

* We use database to connect between all users of the system and store information.

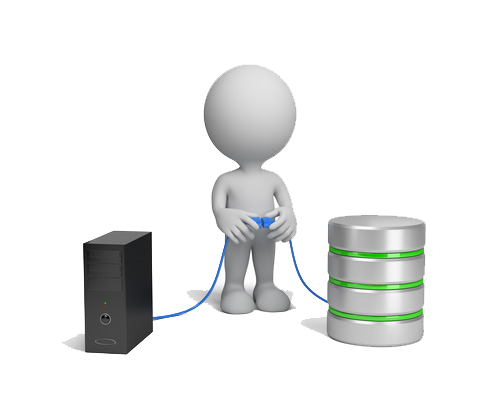


Figure : System Architecture

# System Functional requirements:

## Forgot password:

* If anyone of the users forgot their passwords, they could ask for a new password and the Administrator will replace their old passwords with new ones.

## View users activity table:

* The administrator will have the ability to view the activities of each user to track the system, monitor the behavior of and progress of users, and change passwords solicited by password reset requests.

## View login attempts table:

* The administrator will view the table of login attempts which includes all the requests of users who forgot their passwords and need to change it, or attempts others to breach the system through the login page.

## View users table:

* Only the administrator can view all the users in the system (company).

## Add new user:

* Only the administrator can add a new user to the system by adding specific information (Username – Password – Role – ID).

## Update existing user information:

* The administrator can update user information.

## Create project & assign project to users:

* Only the administrator can create a new project to the system and assign to this project a specific number of users be it developers or testers.

## View project table & analyze project progress:

* The administrator will have the ability to view a table of all projects in the system and analyze their progress by following the dates.

## View bug table:

* The administrator can view a table of bugs that are created by the tester & assign a developer to solve this bug.

## Add bug details (types and severity):

* The administrator can add bug types and severity.

## Reset password:

* Developers have the access to reset their passwords, but the administrator will receive a notification when this action happens.

## View assigned projects table & view the other assigned developers in the same projects:

* **Developers:** The developers can view the list of the assigned projects which have been assigned by the administrator & can also view the other assigned developers in the same project to make the communication between them smooth, easy & simple.
* **Testers:** The testers can view the list of the assigned projects which have been assigned by the administrator & can also view the other assigned users whether developers or testers in the same project.

## View assigned bugs table:

* **Developers & Testers** can view the list of the assigned bugs which have been discovered (in the tester’s case) or assigned by the administrator (in the developer’s case).

## Update bug status:

* **Developers:** The developers can update the assigned bug status from in progress into to be tested after adding the solution of this bug.
* **Testers:** The testers can update the assigned bug status from to be tested into closed after reviewing the solution of the developer.

## Report bug:

* Only testers have the access to create a new bug and add the details of this bug.

## Update bug details:

* Only testers have the access to update bug details.

## Delete bug:

* Only testers have the access to delete the bug.

## View an overview of bugs status in a specific project:

* Only testers have the access to View an overview of bugs status in a specific project in pie chart and cards.

# Interface requirements:

## User interfaces:

Graphical user interface, diagram, application, Teams

Description automatically generated

Figure 4:Intro page

Graphical user interface, application, Teams

Description automatically generated

Figure 5:Login Page

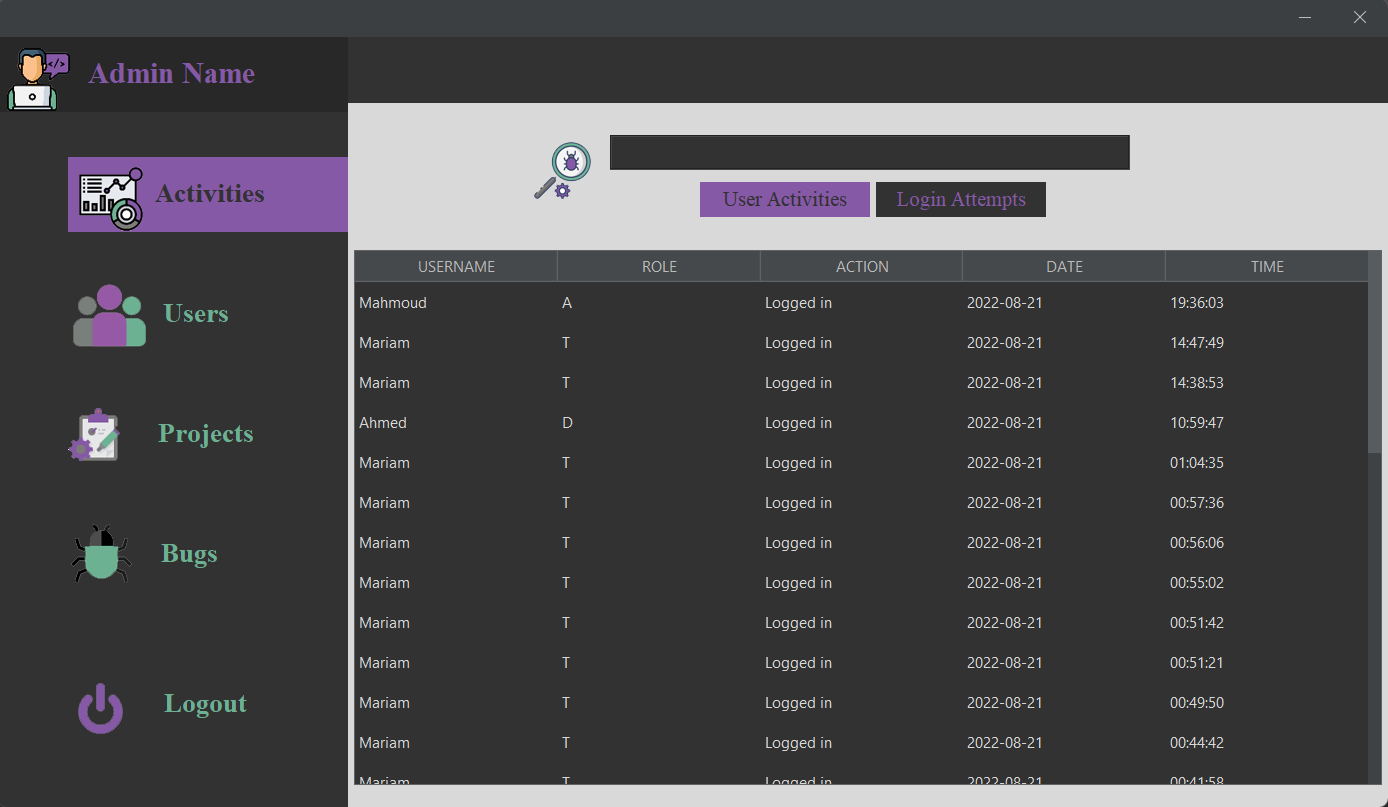


Figure :Admin page: Activity &login attempts table

A screenshot of a computer

Description automatically generated

Figure :Admin page: Users table

Figure 8:Admin Page: Projects table

Figure 9:Admin Page: User information

A screenshot of a computer

Description automatically generatedGraphical user interface, application, Teams

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

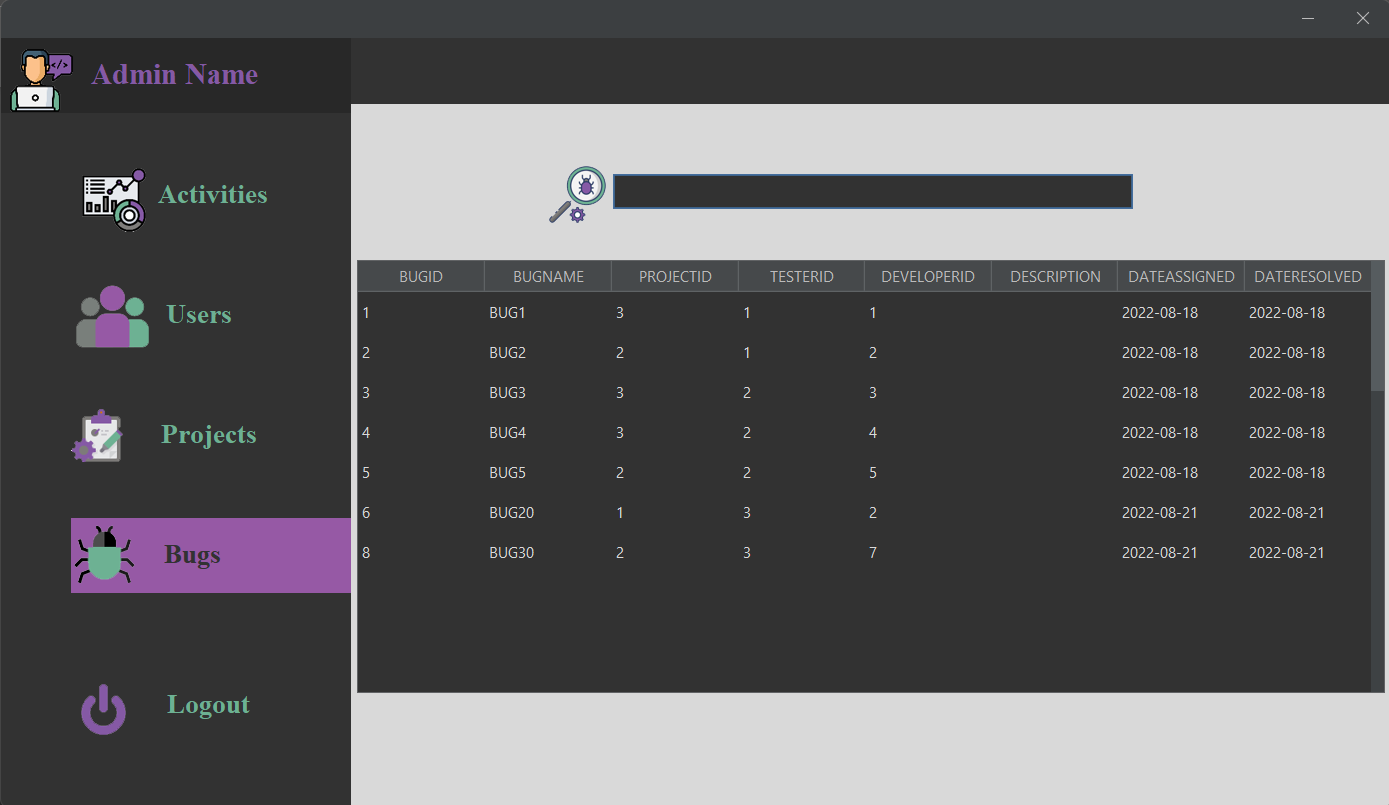


Figure : Admin Page: Bugs Table

A screenshot of a computer

Description automatically generated with medium confidence

Figure :Developer Page: Dashboard

A screenshot of a computer

Description automatically generated with medium confidence

Figure :Developer Page: Projects Table

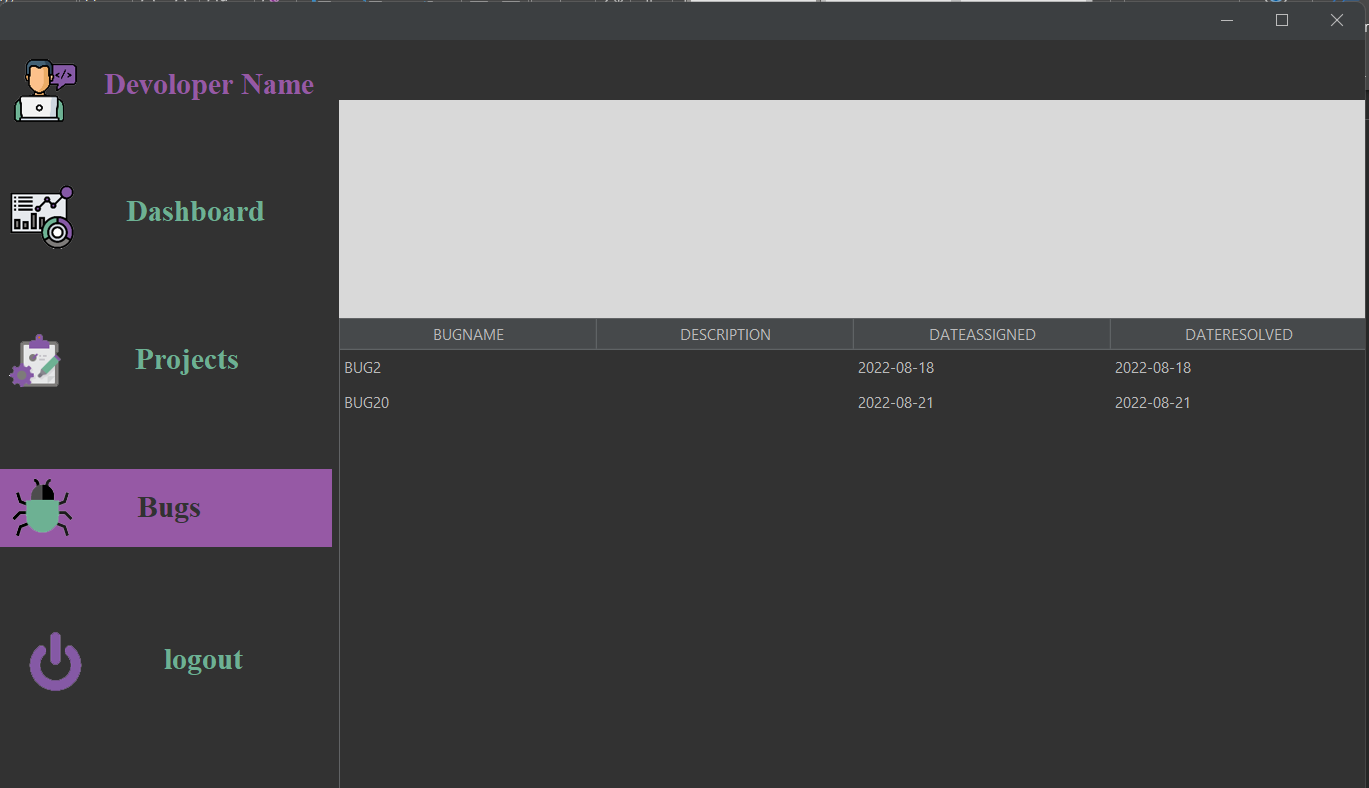


Figure :Developer Page: Bugs Table

A screenshot of a computer

Description automatically generated

Figure :Tester Page: Dashboard

A screenshot of a computer

Description automatically generated

Figure :Tester Page: Projects Table

A screenshot of a computer

Description automatically generated with medium confidence

Figure 16:Tester Page: Assigned people to the same project

A screenshot of a computer

Description automatically generated

Figure :Tester Page: Bugs Panel

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 18:Tester Page: View all the bugs in a specific project

Graphical user interface, application

Description automatically generated

Figure 19:Tester Page: View Bug Details

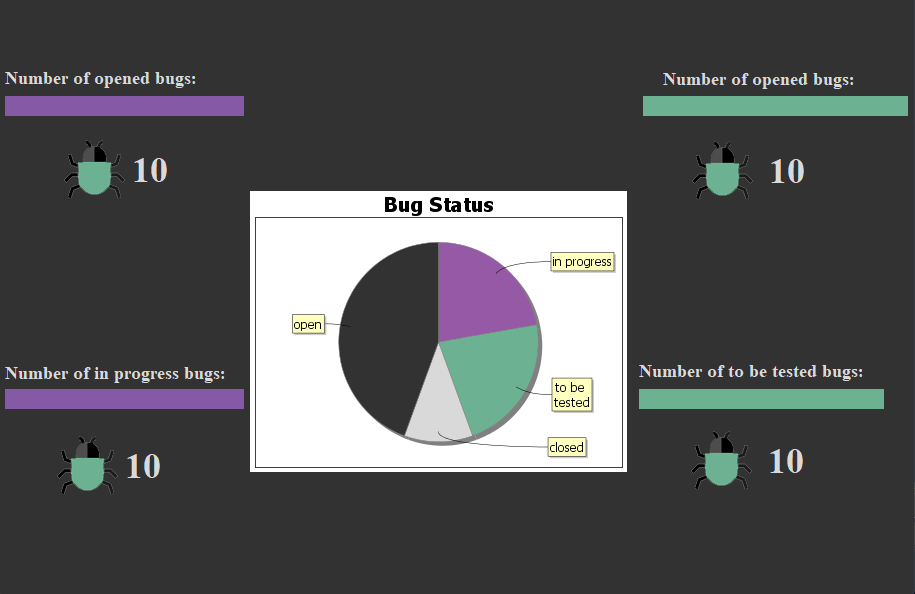


Figure :Tester Page: View Overview of Bugs status

## Software Interfaces:

* Database will be created using apache derby and will be maintained and updated using Git servers.

# Non-functional requirements:

## Availability:

* The system should be available during working hours.

## Security:

* No one can access the system from outside the company.

## Maintainability:

* The system consists of modules every module consists of classes in order, so the system is easy to maintainable.

## Portability:

* The system is a desktop application can run on several operating systems.

## Usability:

* The system is usable for all users (developers & testers).

## Efficiency

* The system is efficient in a lot of software companies because it solves the bugs of the software.

# System Models and Diagrams:

* Context diagram:

A picture containing chart

Description automatically generated

Figure :context diagram

* Use case diagram:
* Class diagram:

# System Evolution:

* The system should be able to work on different operating systems.
* It should work properly on devices with low specifications. Only the server on which it’s installed will be powerful.

# Future Work:

* Administrator:
* Print users data in a table.
* Print entire project’s bug information.
* Print information about each bug in the project individually.
* Developer:
* Add errors and request assistance.
* Print information about each bug in the project individually.
* Enhance the communication between testers and developers by adding comments in each project.

# Time Plan:

## Work Breakdown Structure:

Timeline

Description automatically generated

## Gantt chart:

A picture containing chart

Description automatically generated

# Appendices:

# References:

* <https://projectsgeek.com/2016/02/bug-tracking-system-java-project.html>