

○Prepared by:

- Mahmoud Ehab Mahmoud Abdeen.
- Ahmed Khaled Mohana Tohamy.
- Mariam Mohamed Abdelmonem Ismail.

Under supervision of
DR/Ma¹nal showman

Table of Contents:

Team members:	4
List of Figures:	4
1. Preface:	6
1.1 Document Purpose:	6
1.2 Target Users:	6
1.3 Revision History:	6
2. Introduction:	7
2.1 Purpose:	7
2.2 Scope:	7
2.3 Overview:	8
3. Glossary:	8
3.1 Acronyms, definitions, and abbreviations:	8
4. System Users:	9
4.1 System stakeholders:	9
4.2 Users objectives:	9
5. User Requirements definitions:	10
5.1 System Functions:	10
5.2 Constraints:	11
6. System Architecture:	11
7. System Functional requirements:	11
7.1 Forgot password:	12
7.2 View users activity table:	12
7.3 View login attempts table:	12
7.4 View users table:	12
7.5 Add new user:	12
7.6 Update existing user information:	12
7.7 Create project & assign project to users:	12
7.8 View project table & analyze project progress:	12
7.9 View bug table:	12
7.10 Add bug details (types and severity):	12
7.11 Reset password:	12

7.12	View assigned projects table & view the other assigned developers in the same projects:.....	12
7.13	View assigned bugs table:.....	13
7.14	Update bug status:.....	13
7.15	Report bug:	13
7.16	Update bug details:	13
7.17	Delete bug:	13
7.18	View an overview of bugs status in a specific project:.....	13
8.	Interface requirements:.....	13
8.1	User interfaces:	13
8.2	Software Interfaces:	23
9.	Non-functional requirements:	24
9.1	Availability:	24
9.2	Security:	24
9.3	Maintainability:.....	24
9.4	Portability:	24
9.5	Usability:.....	24
9.6	Efficiency	24
10.	System Models and Diagrams:.....	24
11.	System Evolution:	26
12.	Future Work:	26
13.	Time Plan:	27
13.1	Work Breakdown Structure:	27
13.2	Gantt chart:	27
14.	References:	28

Team members:

Name	Code	Work in SRS
Ahmed Khaled Mohana Tohamy.	20026	<ul style="list-style-type: none"> ○ Introduction. ○ System architecture. ○ System non-functional requirements. ○ System Models and diagrams (use case diagram). ○ Glossary.
Mahmoud Ehab Mahmoud Abdeen.	20010	<ul style="list-style-type: none"> ○ Preface. ○ User requirements definitions. ○ Interface requirements. ○ System Models and diagrams (Class diagram). ○ Glossary.
Mariam Mohamed Abdelmonem Ismail	20047	<ul style="list-style-type: none"> ○ System users. ○ System functional requirements. ○ System evolution. ○ Future work. ○ Work Plan. ○ System Models and diagrams (Context diagram). ○ Glossary.

List of Figures:

Figure 1 : System Modules hierarchy	7
Figure 2 : System Functions hierarchy.....	8
Figure 3 : System Architecture.....	11
Figure 4 : Loading page.....	13
Figure 5 : Login Page	14
Figure 6 : Admin page: Activity &login attempts table.....	14
Figure 7 : Admin page: Users table.....	15
Figure 8 : Admin Page: User information	15
Figure 9:Admin Page: Projects table.....	16
Figure 10 : Admin Page: Project Information	16
Figure 11: Admin Page: Bugs Table	17
Figure 12 : Rating page	17
Figure 13 : Developer Page: Dashboard	18
Figure 14 : Developer Page: Projects Table.....	18
Figure 15 : Developer Page: Bugs Table	19

Figure 16 : Tester Page: Dashboard	19
Figure 17: Tester Page: Bug Information.....	20
Figure 18 : Tester Page: Project Table	20
Figure 19:Tester Page: Assigned people to the same project	21
Figure 20:Tester Page: Bugs Panel	21
Figure 21:Tester Page: View all the bugs in a specific project.....	22
Figure 22:Tester Page: View Bug Details or add new bug	22
Figure 23:Tester Page: View Overview of Bugs Status.....	23
Figure 24:Rating Page	23
Figure 25:context diagram.....	24
Figure 26 : Class diagram.....	26
Figure 27 : Use case diagram	26
Figure 28 : Work Break Down Structure	27
Figure 29 : Gannt chart.....	27
Figure 30 : Updated Gannt Chart	28

1. Preface:

1.1 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.

1.2 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.

1.3 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 is added	12-8-2022
0.6	Mahmoud Ehab	Minor changes in system functional requirements	21-8-2022
0.7	Mahmoud Ehab	Class diagram is updated	21-8-2022
0.8	Ahmed Khaled	Use case diagram is updated	21-8-2022

0.9	Ahmed Khaled	Small change in stakeholders	2-9-2022
1.0	Mariam Mohamed	User interface is updated	2-9-2022

2. Introduction:

2.1 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.

2.2 Scope:

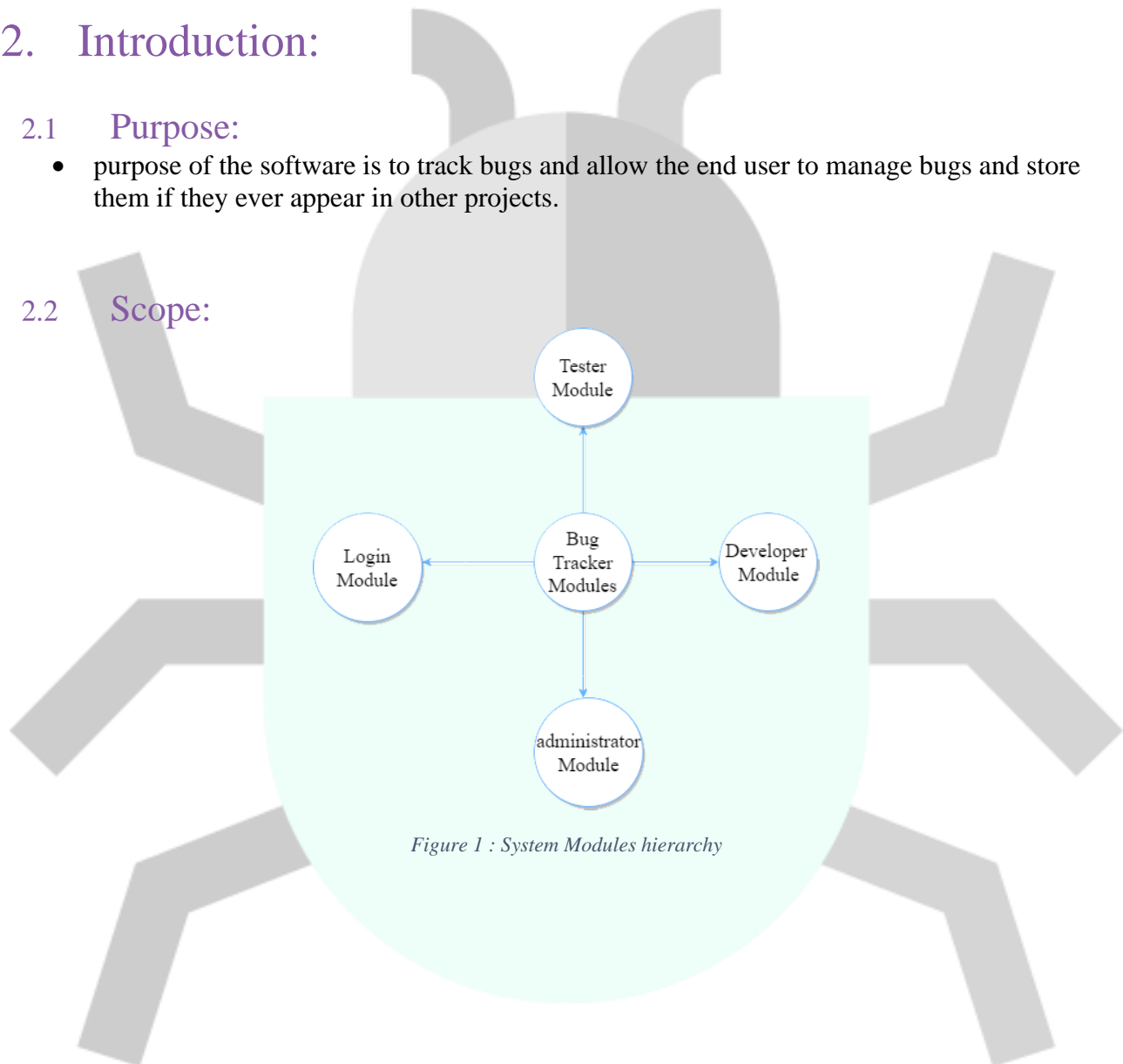


Figure 1 : System Modules hierarchy

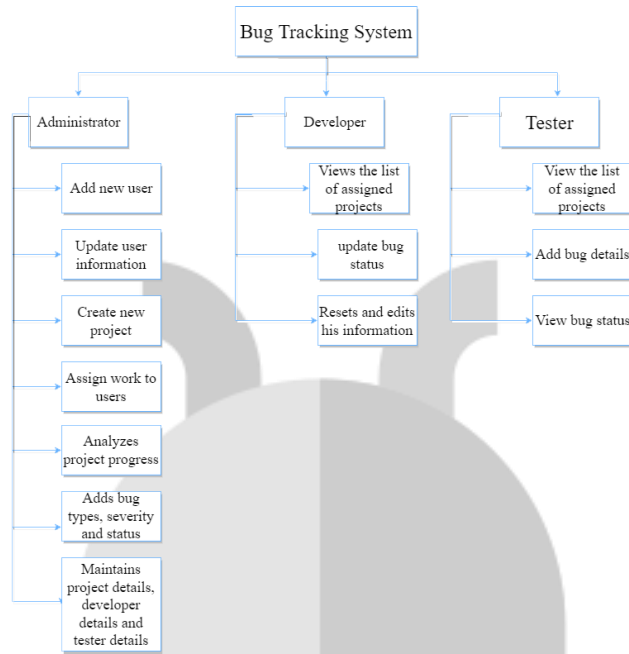


Figure 2 : System Functions hierarchy

2.3 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.

3. Glossary:

3.1 Acronyms, definitions, and abbreviations:

- **FOEHU:** Faculty of Engineering, Helwan University.
- **BTS:** Bug Tracker System.

- **User:** Developer or tester who uses the application.

4. System Users:

4.1 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.

4.2 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.

5. User Requirements definitions:

5.1 System Functions:

- **Login:**
 1. Forgot password.
- **Administrator:**
 2. View user activities table.
 3. View login attempts table.
 4. View users table.
 5. Add new user.
 6. Update existing user information.
 7. Delete users.
 8. View projects table.
 9. Create projects & Assign project to user.
 10. View bugs table.
 11. Add bug details (types and severity).
- **Developer:**
 12. Reset password.
 13. View assigned projects table & view the other assigned developers in the same projects.
 14. View assigned bugs table.
 15. Update bug status.
- **Tester:**

16. View assigned projects table.
17. View users of each project assigned to him.
18. View assigned bugs table.
19. Report bug.
20. Update bug details.
21. Update bug status.
22. Delete bug.
23. View an overview of bugs status in specific project.

5.2 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.

6. System Architecture:

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



Figure 3 : System Architecture

7. System Functional requirements:

7.1 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.

7.2 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.

7.3 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.

7.4 View users table:

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

7.5 Add new user:

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

7.6 Update existing user information:

- The administrator can update user information.

7.7 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.

7.8 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.

7.9 View bug table:

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

7.10 Add bug details (types and severity):

- The administrator can add bug types and severity.

7.11 Reset password:

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

7.12 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.

7.13 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).

7.14 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.

7.15 Report bug:

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

7.16 Update bug details:

- Only testers have the access to update bug details.

7.17 Delete bug:

- Only testers have the access to delete the bug.

7.18 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.

8. Interface requirements:

8.1 User interfaces:

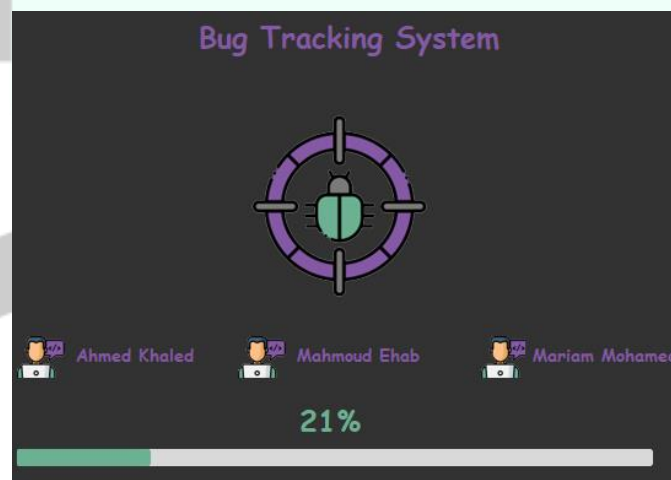


Figure 4 : Loading page

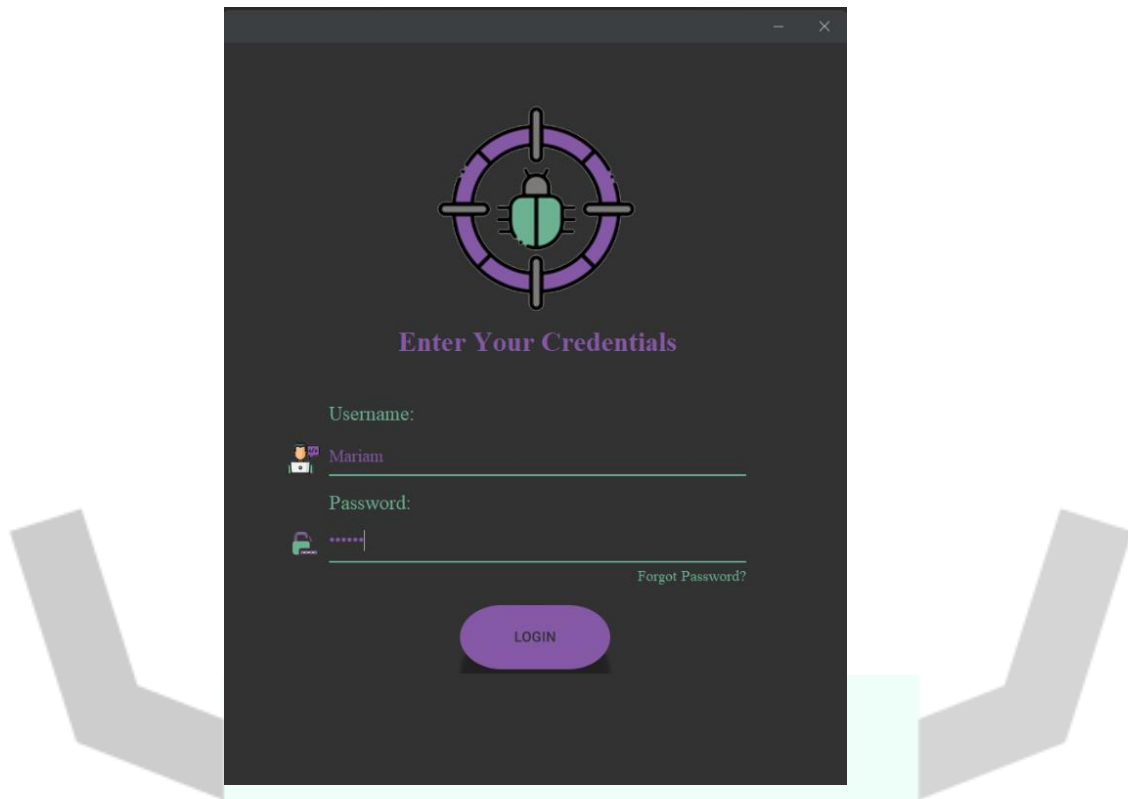


Figure 5 : Login Page

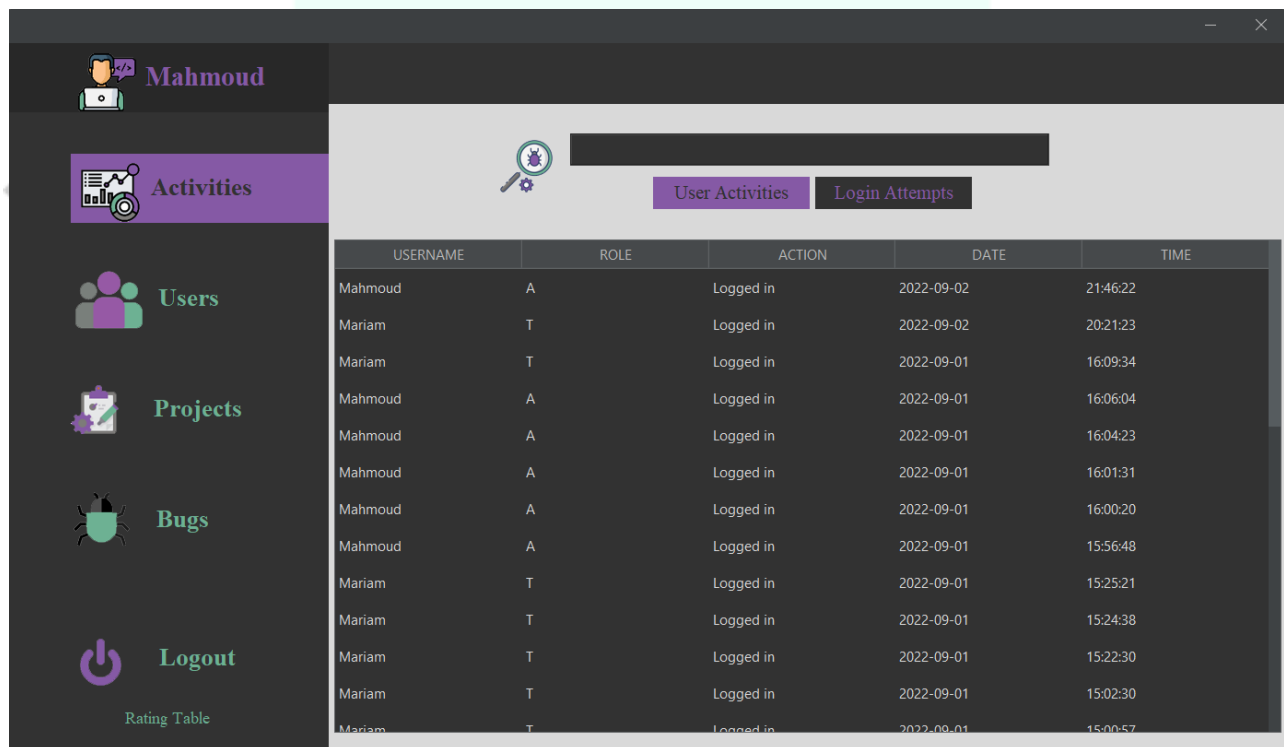


Figure 6 : Admin page: Activity & login attempts table

Figure 7 : Admin page: Users table

User Information

ID

Username

Password

Role

Confirm **Cancel**

User Information

User ID **Edit**

Username

Password

Role

Confirm **Delete** **Cancel**

Figure 8 : Admin Page: User information

Mahmoud
Activities
Users
Projects
Bugs
Logout

Rating Table

Update

USERID	USERNAME	PASSWORD	ROLE	CREATIONDATE	CREATIONTIME
1	Mahmoud	Admin	A	2022-08-18	16:43:06
2	Ahmed	Developer	D	2022-08-18	16:43:06
3	Mariam	Tester	T	2022-08-18	16:43:06
4	Maged	Tester	T	2022-08-23	18:53:42
5	Arnold	1235	D	2022-08-26	18:27:05
6	Khaled	Admin	A	2022-08-23	19:55:31
8	Marawan	qwerty	T	2022-08-26	21:39:39

Add User **Delete All**

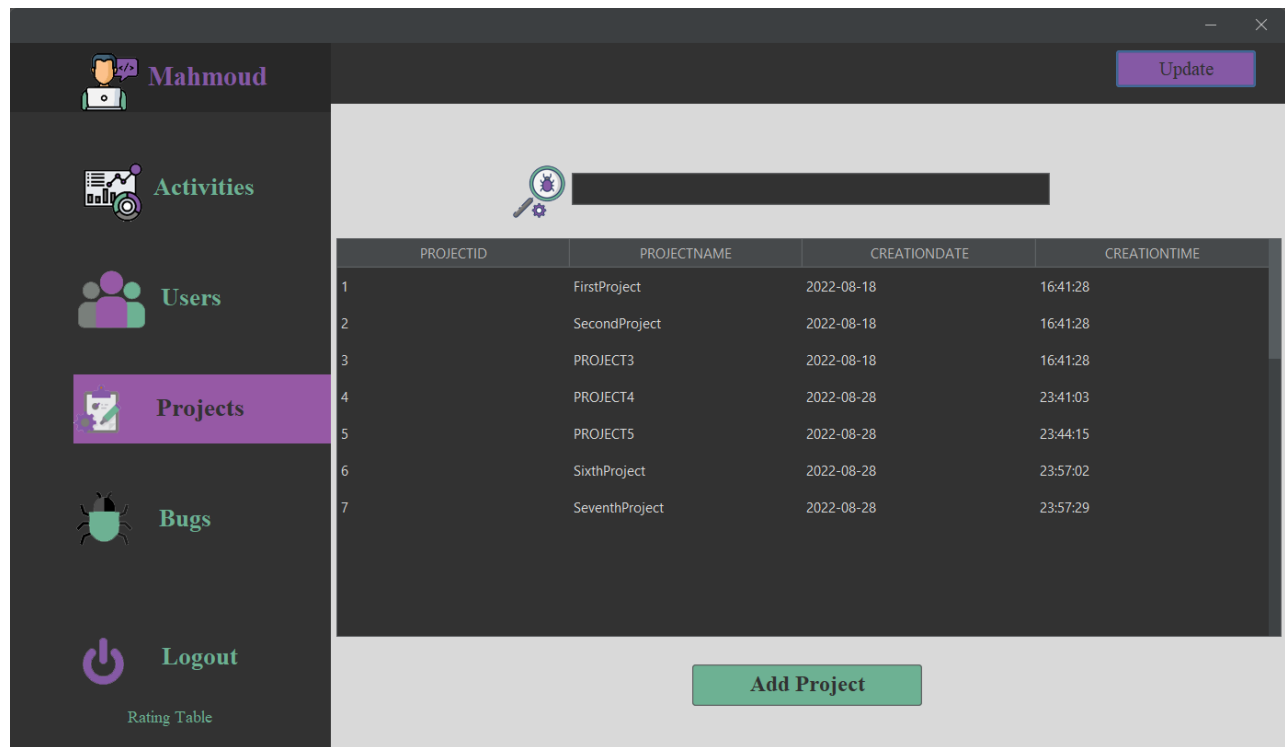


Figure 9:Admin Page: Projects table

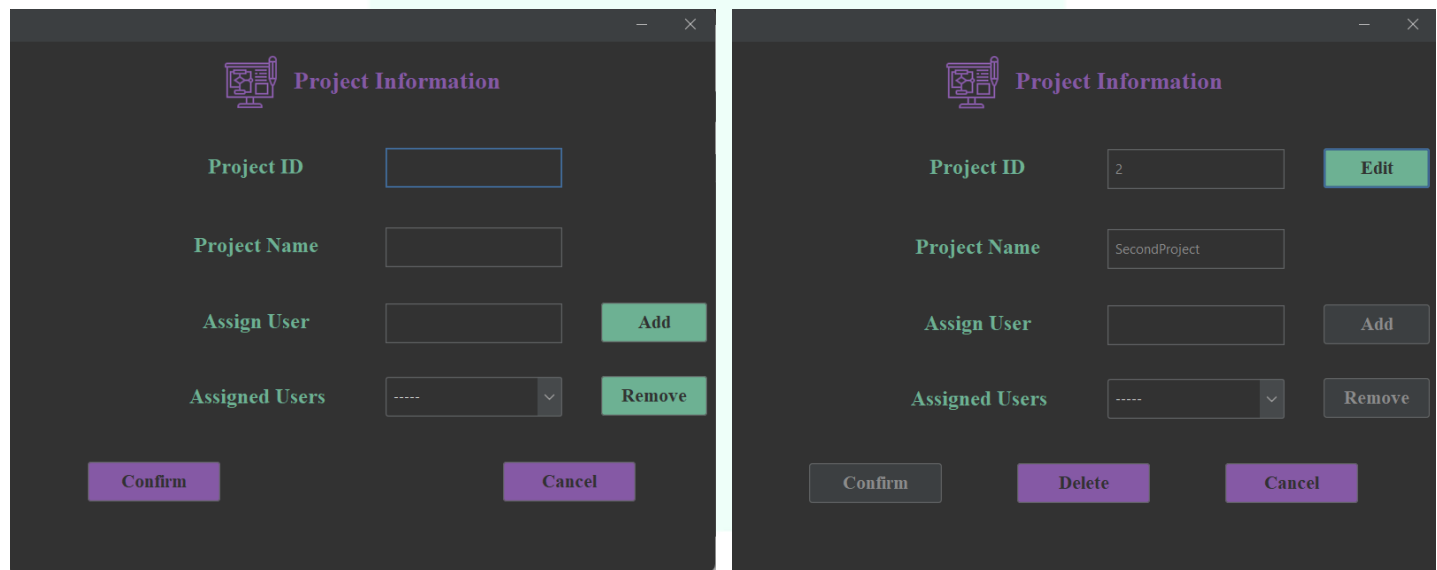
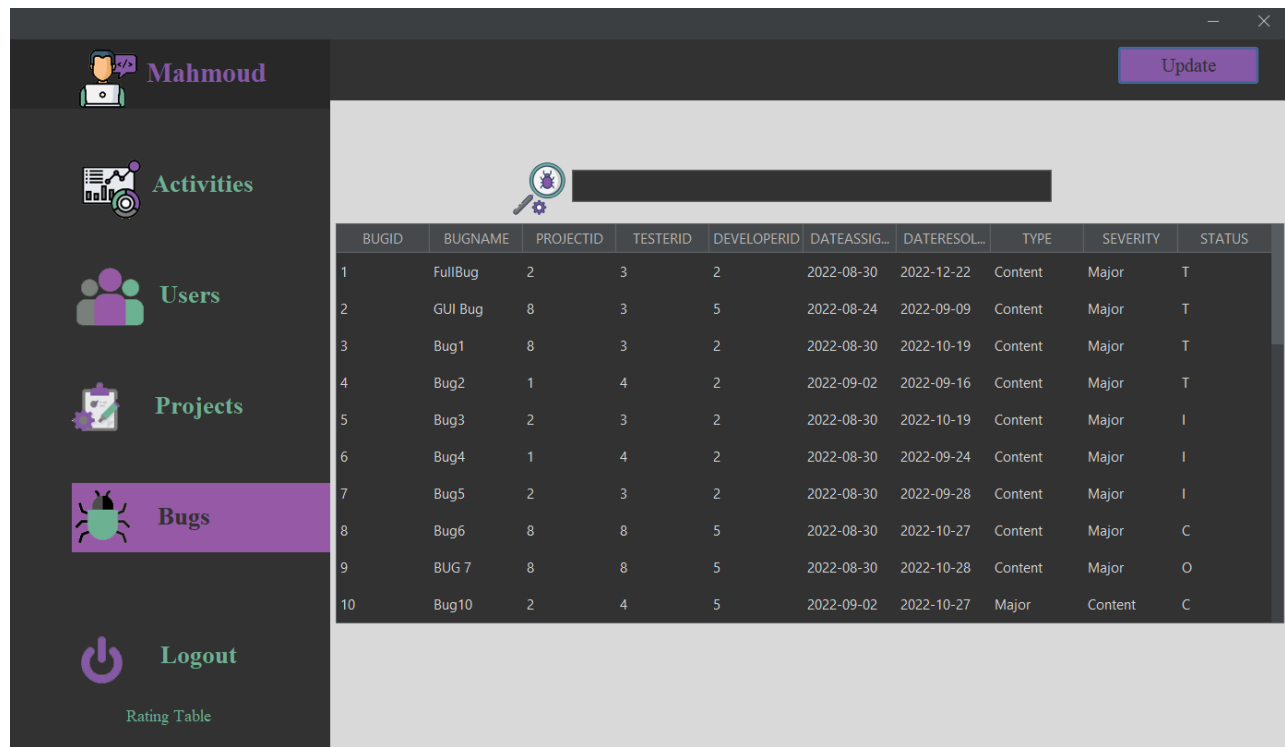


Figure 10 : Admin Page: Project Information



BUGID	BUGNAME	PROJECTID	TESTERID	DEVELOPERID	DATEASSIG...	DATERESOL...	TYPE	SEVERITY	STATUS
1	FullBug	2	3	2	2022-08-30	2022-12-22	Content	Major	T
2	GUI Bug	8	3	5	2022-08-24	2022-09-09	Content	Major	T
3	Bug1	8	3	2	2022-08-30	2022-10-19	Content	Major	T
4	Bug2	1	4	2	2022-09-02	2022-09-16	Content	Major	T
5	Bug3	2	3	2	2022-08-30	2022-10-19	Content	Major	I
6	Bug4	1	4	2	2022-08-30	2022-09-24	Content	Major	I
7	Bug5	2	3	2	2022-08-30	2022-09-28	Content	Major	I
8	Bug6	8	8	5	2022-08-30	2022-10-27	Content	Major	C
9	BUG 7	8	8	5	2022-08-30	2022-10-28	Content	Major	O
10	Bug10	2	4	5	2022-09-02	2022-10-27	Major	Content	C

Figure 11: Admin Page: Bugs Table



ID	USERNAME	ROLE	RATE
1	Mahmoud	A	5
2	Mariam	T	3
3	Mahmoud	A	3
4	Mariam	T	4

Figure 12 : Rating page

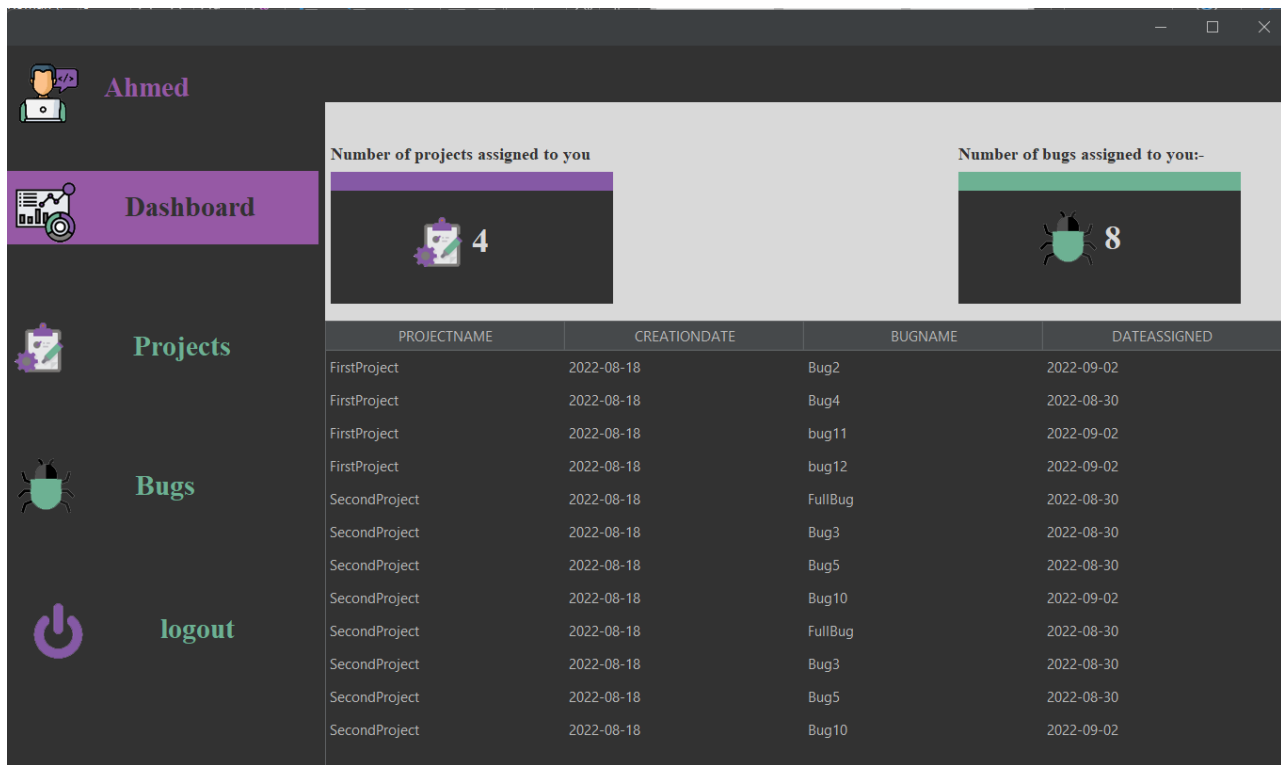


Figure 13 : Developer Page: Dashboard

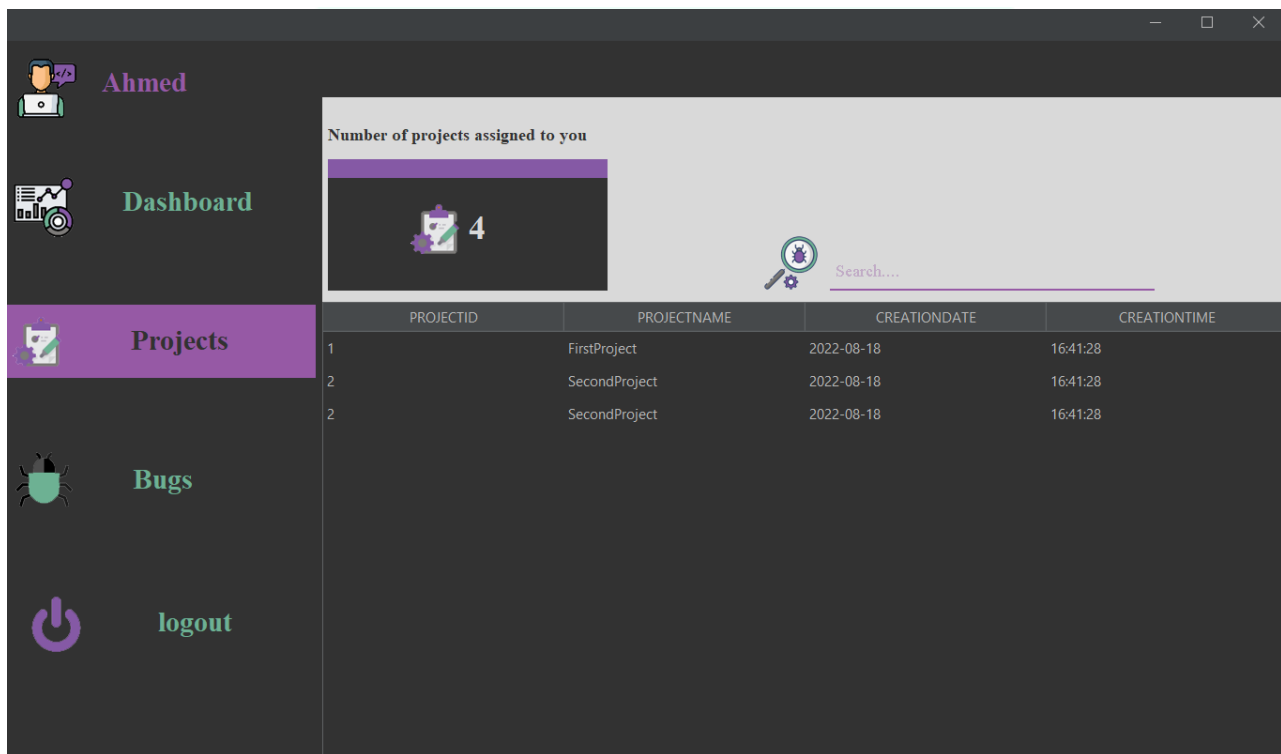
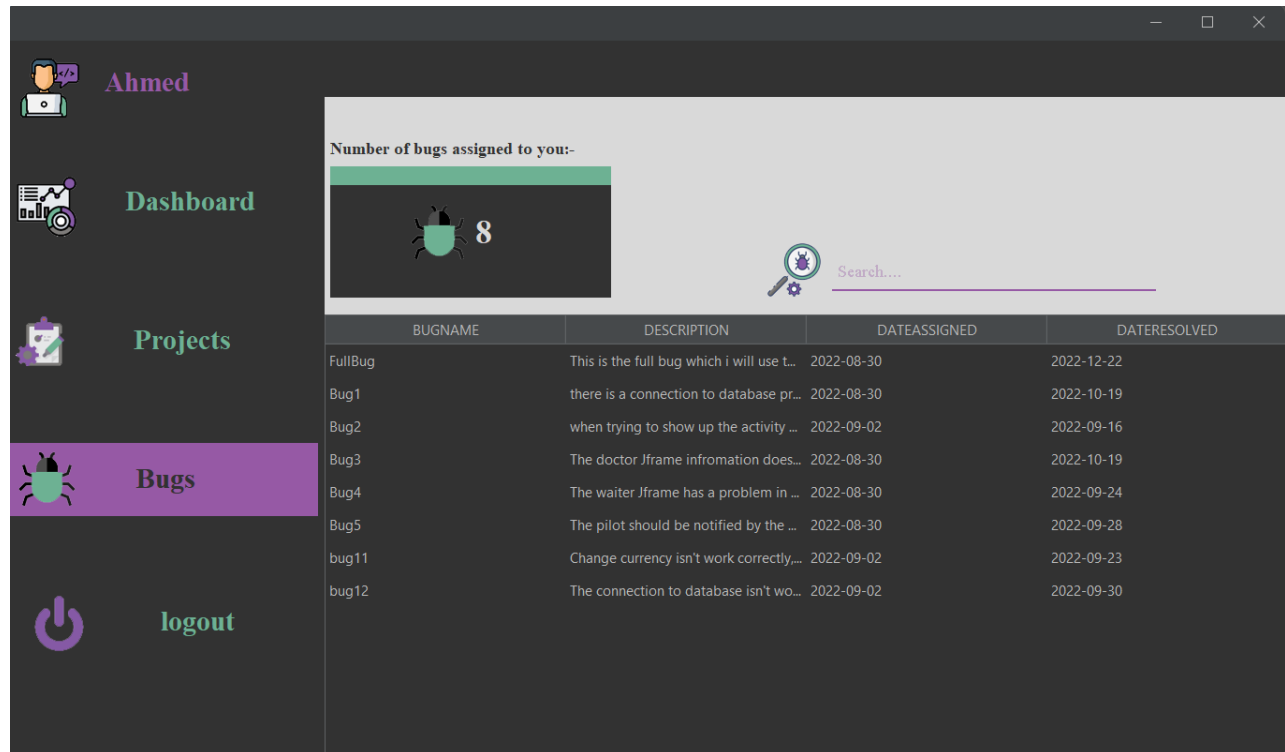


Figure 14 : Developer Page: Projects Table



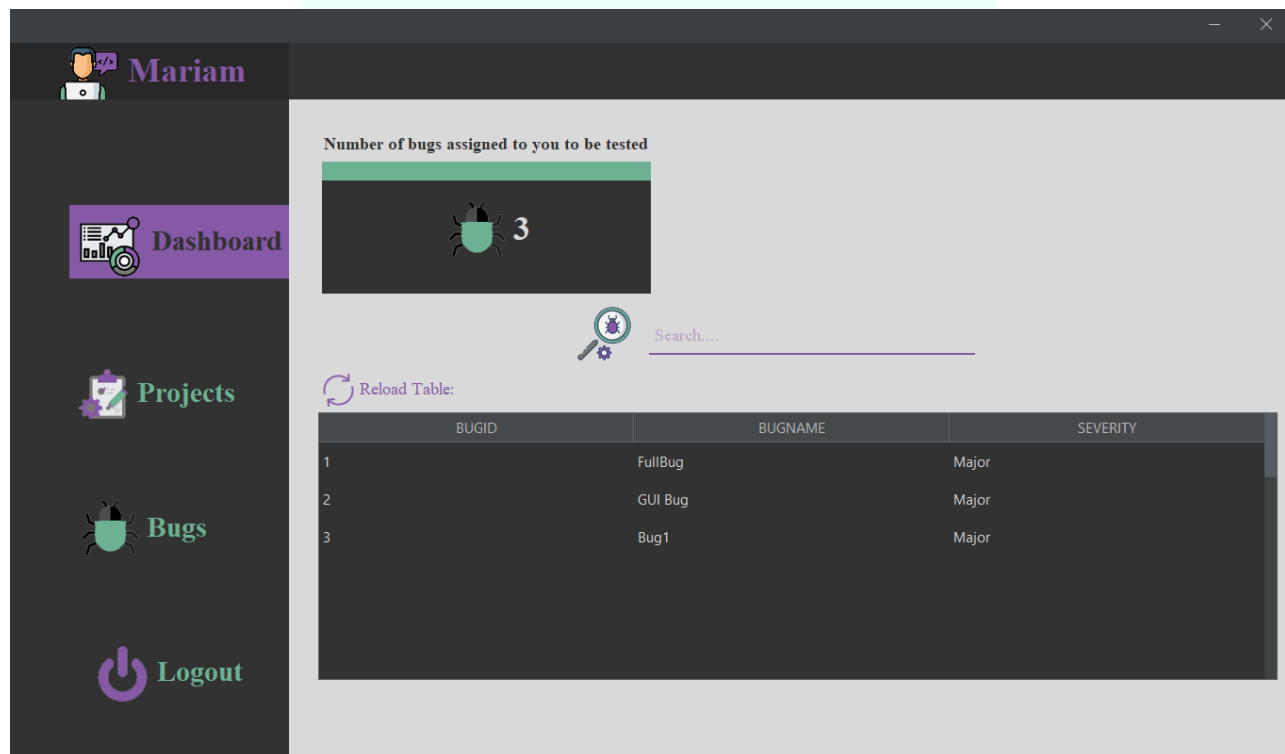
Number of bugs assigned to you:-

8

Search....

BUGNAME	DESCRIPTION	DATEASSIGNED	DATERESOLVED
FullBug	This is the full bug which i will use t...	2022-08-30	2022-12-22
Bug1	there is a connection to database pr...	2022-08-30	2022-10-19
Bug2	when trying to show up the activity ...	2022-09-02	2022-09-16
Bug3	The doctor Jframe infromation does...	2022-08-30	2022-10-19
Bug4	The waiter Jframe has a problem in ...	2022-08-30	2022-09-24
Bug5	The pilot should be notified by the ...	2022-08-30	2022-09-28
bug11	Change currency isn't work correctly,...	2022-09-02	2022-09-23
bug12	The connection to database isn't wo...	2022-09-02	2022-09-30

Figure 15 : Developer Page: Bugs Table



Number of bugs assigned to you to be tested

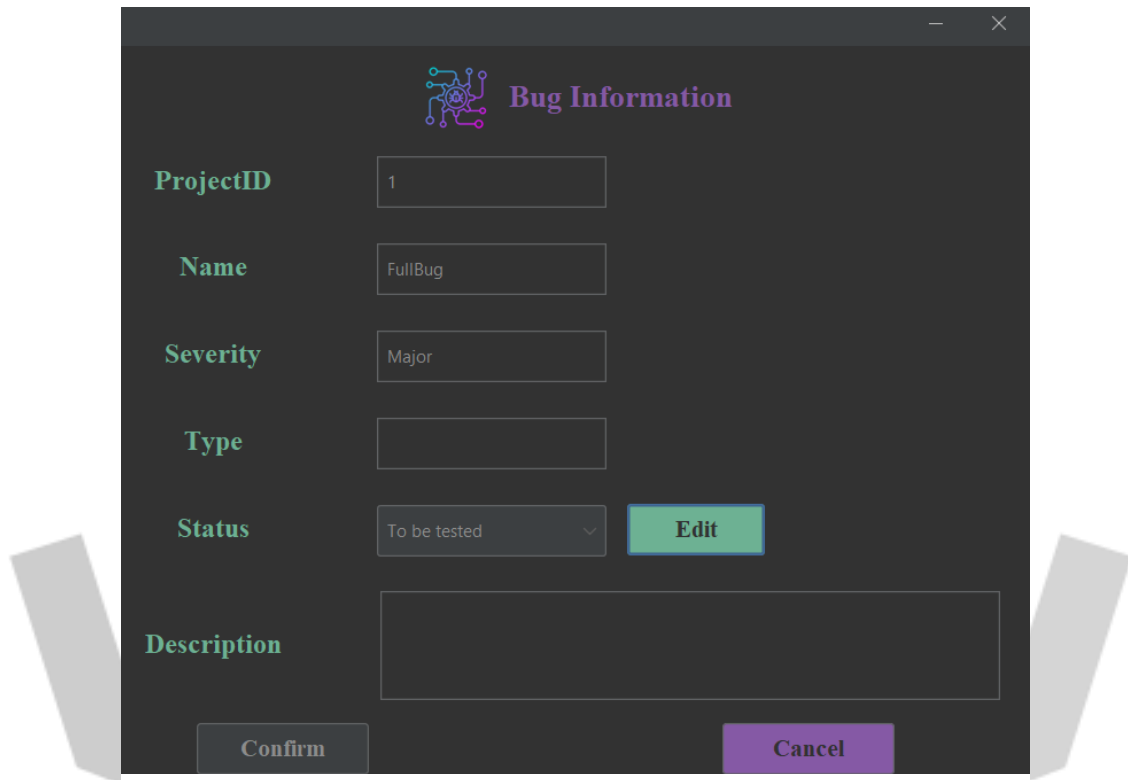
3

Search....

Reload Table:

BUGID	BUGNAME	SEVERITY
1	FullBug	Major
2	GUI Bug	Major
3	Bug1	Major

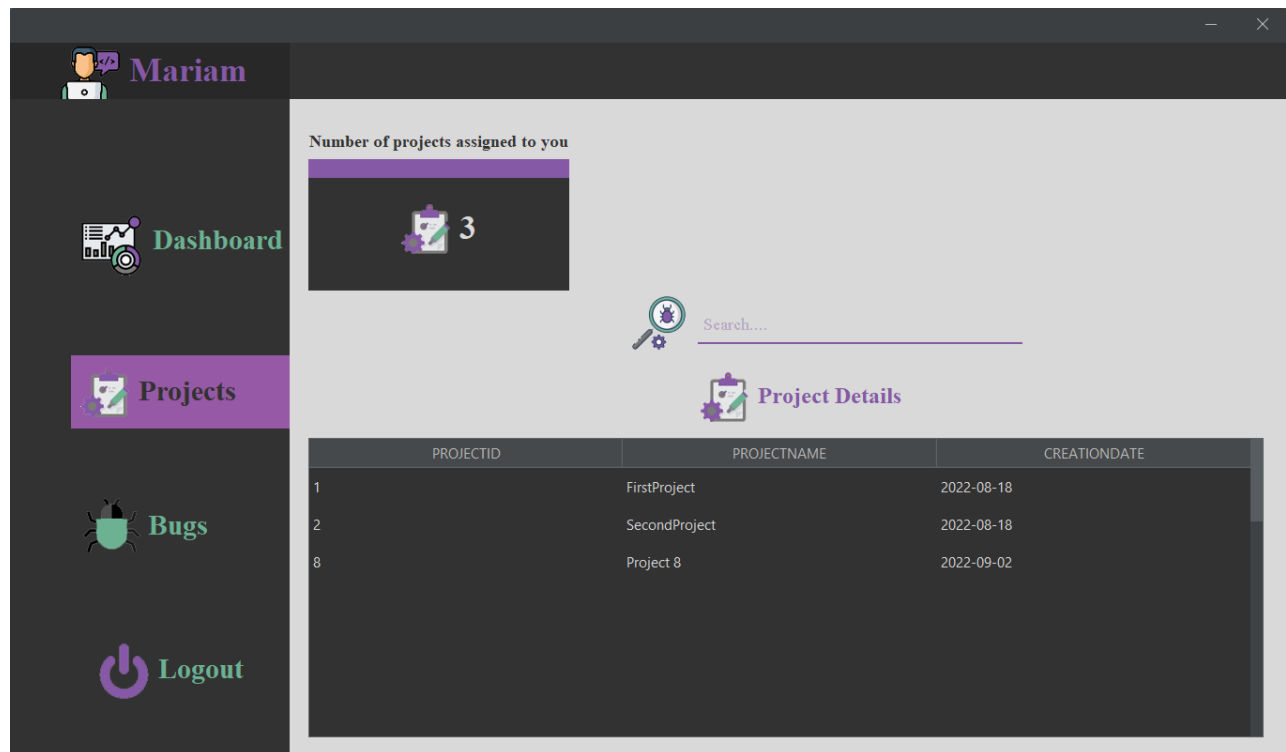
Figure 16 : Tester Page: Dashboard



A screenshot of a 'Bug Information' form. The form has a dark background with light-colored text and input fields. It includes fields for ProjectID (1), Name (FullBug), Severity (Major), Type (empty), Status (To be tested), and Description (empty). There are 'Edit', 'Confirm', and 'Cancel' buttons.

Field	Value
ProjectID	1
Name	FullBug
Severity	Major
Type	
Status	To be tested
Description	

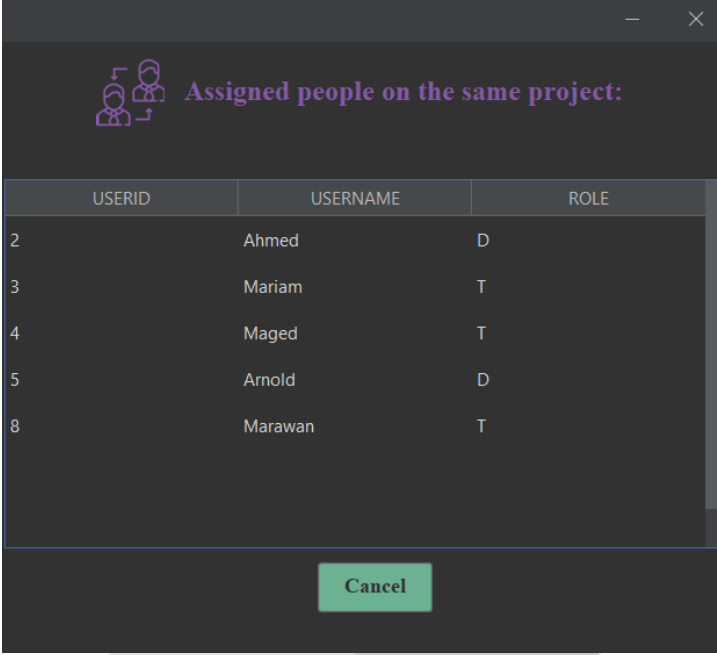
Figure 17: Tester Page: Bug Information



A screenshot of a 'Tester Page' dashboard. The left sidebar contains navigation links: Mariam, Dashboard, Projects, Bugs, and Logout. The main content area shows the number of projects assigned (3) and a search bar. Below the search bar is a 'Project Details' link and a table of projects.

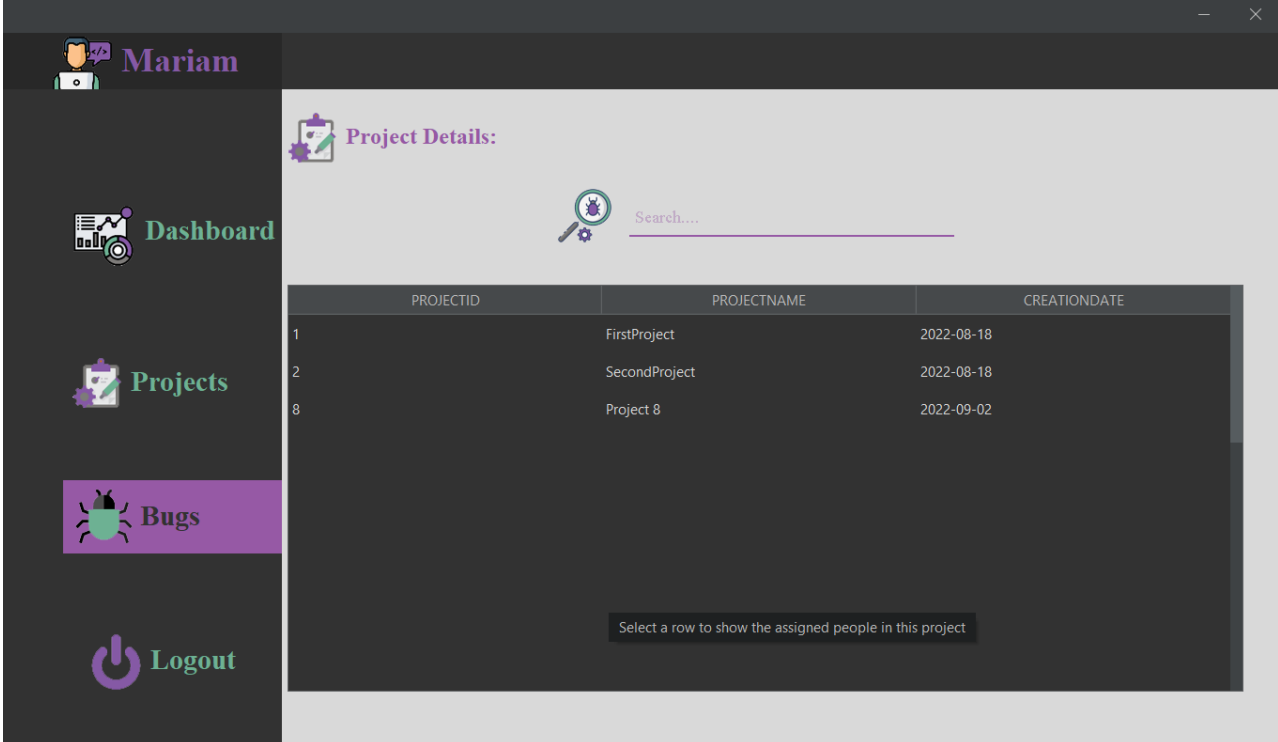
PROJECTID	PROJECTNAME	CREATIONDATE
1	FirstProject	2022-08-18
2	SecondProject	2022-08-18
8	Project 8	2022-09-02

Figure 18 : Tester Page: Project Table



USERID	USERNAME	ROLE
2	Ahmed	D
3	Mariam	T
4	Maged	T
5	Arnold	D
8	Marawan	T

Figure 19:Tester Page: Assigned people to the same project



Mariam

Project Details:

Search....

PROJECTID	PROJECTNAME	CREATIONDATE
1	FirstProject	2022-08-18
2	SecondProject	2022-08-18
8	Project 8	2022-09-02

Select a row to show the assigned people in this project

Figure 20:Tester Page: Bugs Panel

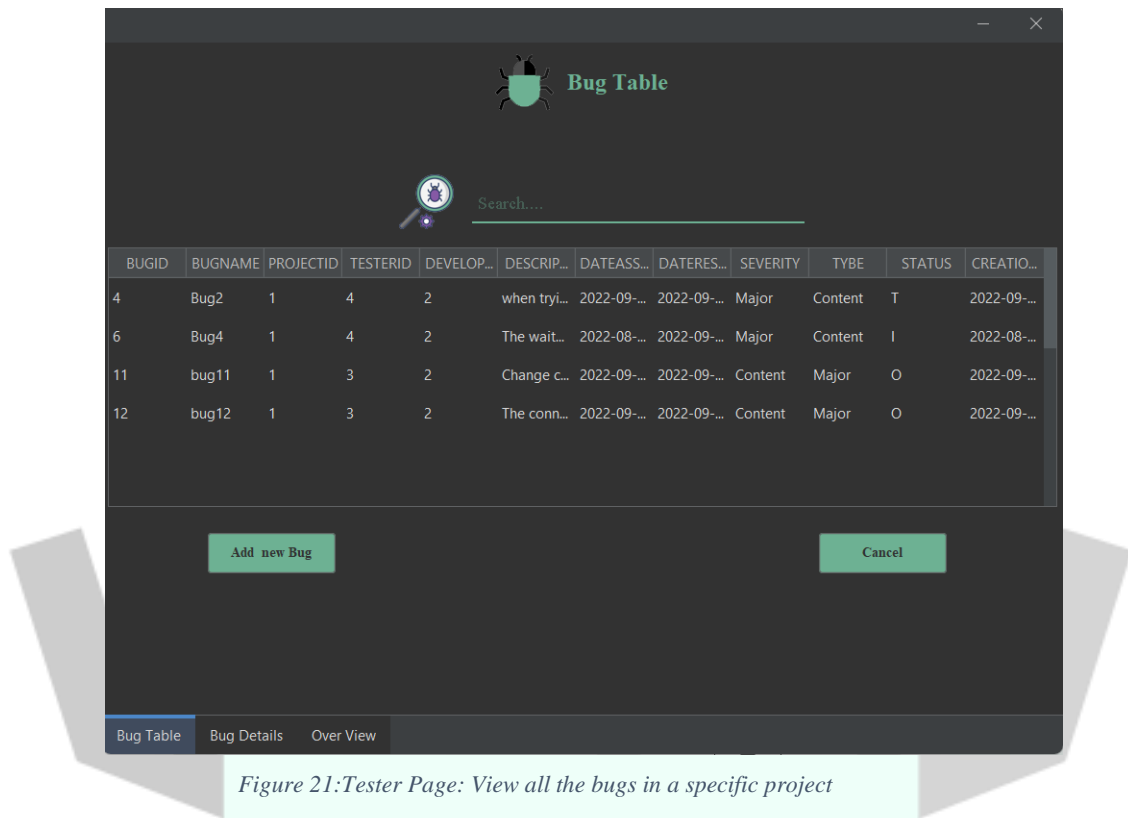


Figure 21:Tester Page: View all the bugs in a specific project

The screenshot shows a window titled "Bug Details:" with a bug icon. It contains the following fields and controls:

- Project ID:** Text input with value "1".
- Status:** Dropdown menu with value "To be tested".
- Bug Name:** Text input with value "Bug2".
- Severity:** Dropdown menu with value "Major".
- Reporter Name:** Text input with value "Maged".
- Type:** Dropdown menu with value "Content".
- Developer Name:** Text input with value "Ahmed".
- Description:** Text area with value "show up the activity of my credit card , it shows only the activity of today not the the activity of the whole month".

At the bottom are three buttons: "Confirm", "Delete", and "Cancel". At the very bottom is a tab bar with three tabs: "Bug Table", "Bug Details", and "Over View". The "Bug Details" tab is currently selected.

The screenshot shows a window titled "Bug Details:" with a bug icon. It contains the following fields and controls:

- Project ID:** Text input with value "1".
- Status:** Dropdown menu with value "Open".
- Bug Name:** Empty text input field.
- Severity:** Dropdown menu with value "-----".
- Reporter Name:** Text input with value "Mariam".
- Type:** Dropdown menu with value "-----".
- Developer Name:** Empty text input field.
- Description:** Empty text area.

At the bottom are two buttons: "Confirm" and "Cancel". At the very bottom is a tab bar with three tabs: "Bug Table", "Bug Details", and "Over View". The "Bug Details" tab is currently selected.

Figure 22:Tester Page: View Bug Details or add new bug

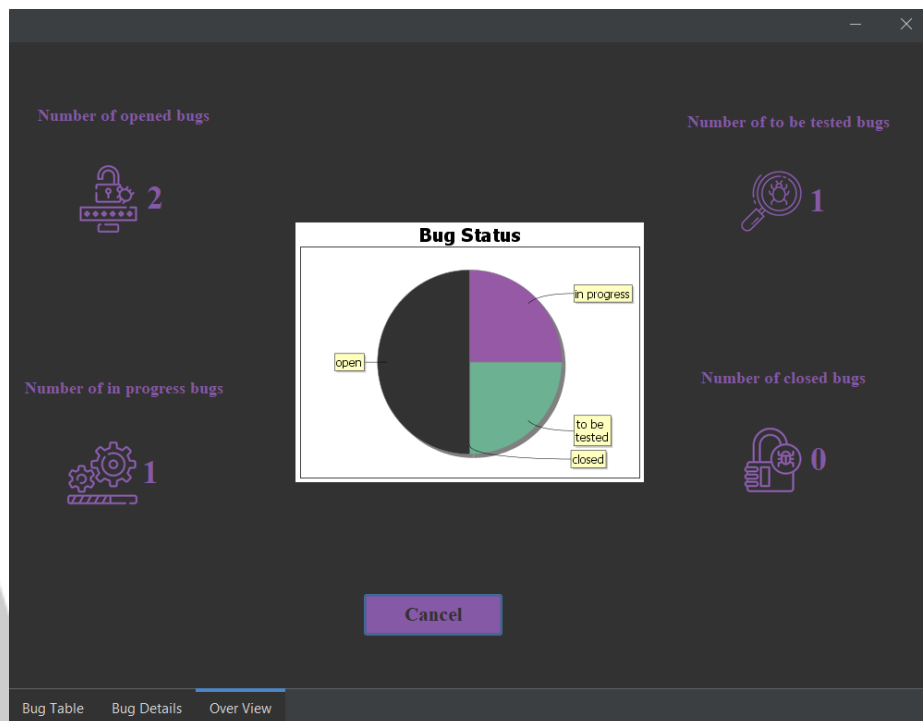


Figure 23: Tester Page: View Overview of Bugs Status

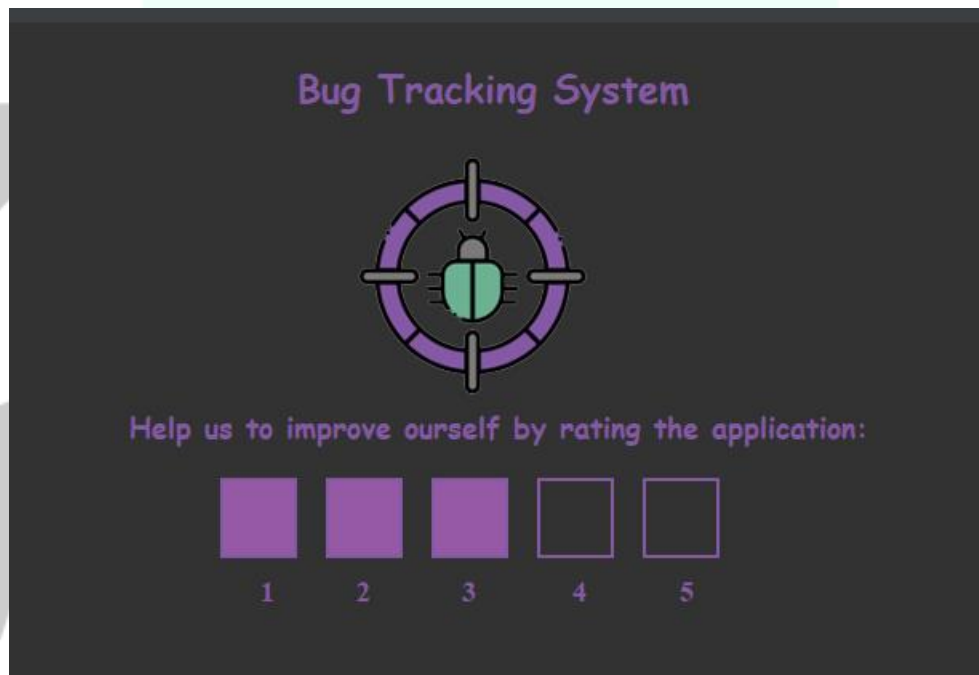


Figure 24: Rating Page

8.2 Software Interfaces:

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

9. Non-functional requirements:

9.1 Availability:

- The system should be available during working hours.

9.2 Security:

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

9.3 Maintainability:

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

9.4 Portability:

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

9.5 Usability:

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

9.6 Efficiency

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

10. System Models and Diagrams:

- Context diagram:

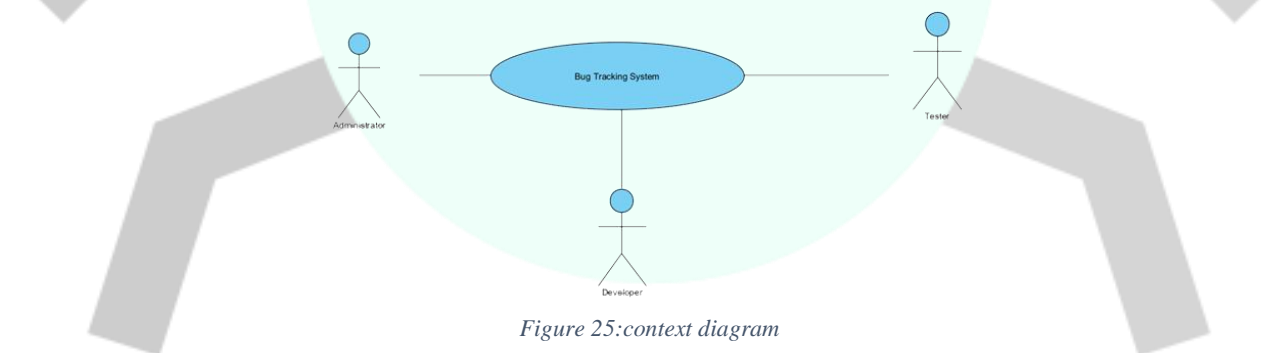


Figure 25:context diagram

- Class diagram:


```

classDiagram
    class Admin {
        +role: String
        +role: String
        +id: int
        +currentUser: String, role: String, d: int
    }
    class Tester {
        +role: String
        +role: String
        +id: int
        +currentUser: String, role: String, d: int
    }
    class Developer {
        +role: String
        +role: String
        +id: int
        +currentUser: String, role: String, d: int
    }
    class Bug {
        +id: int
        +title: String
        +description: String
        +status: String
        +priority: String
        +assignedTo: String
        +createdBy: String
        +createdDate: String
        +lastModifiedBy: String
        +lastModifiedDate: String
    }
    class User {
        +id: int
        +username: String
        +password: String
        +email: String
        +phone: String
        +address: String
        +city: String
        +country: String
        +zip: String
        +gender: String
        +maritalStatus: String
        +education: String
        +occupation: String
        +company: String
        +industry: String
        +sector: String
        +subSector: String
        +jobTitle: String
        +jobDescription: String
        +jobLocation: String
        +jobType: String
        +jobStatus: String
        +jobStartDate: String
        +jobEndDate: String
        +jobDuration: String
        +jobExperience: String
        +jobSkills: String
        +jobLanguages: String
        +jobTools: String
        +jobFrameworks: String
        +jobLibraries: String
        +jobPackages: String
        +jobPlugins: String
        +jobExtensions: String
        +jobThemes: String
        +jobFonts: String
        +jobIcons: String
        +jobImages: String
        +jobVideos: String
        +jobAudio: String
        +jobAnimations: String
        +jobEffects: String
        +jobTransitions: String
        +jobFilters: String
        +jobSorters: String
        +jobGroupers: String
        +jobFilters: String
        +jobSorters: String
        +jobGroupers: String
    }
    class Project {
        +id: int
        +name: String
        +description: String
        +status: String
        +priority: String
        +assignedTo: String
        +createdBy: String
        +createdDate: String
        +lastModifiedBy: String
        +lastModifiedDate: String
    }
    class BugTrackingSystem {
        +role: String
        +role: String
        +id: int
        +currentUser: String, role: String, d: int
    }
    Admin --> Bug
    Admin --> User
    Admin --> Project
    Admin --> BugTrackingSystem
    Tester --> Bug
    Tester --> User
    Tester --> Project
    Tester --> BugTrackingSystem
    Developer --> Bug
    Developer --> User
    Developer --> Project
    Developer --> BugTrackingSystem
    Bug --> User
    Bug --> Project
    Bug --> BugTrackingSystem
    User --> BugTrackingSystem
    Project --> BugTrackingSystem
    BugTrackingSystem --> BugTrackingSystem
  
```

The diagram illustrates the roles and interactions within a Bug Tracking System. The main roles are Admin, Tester, and Developer, each with a set of associated use cases. The Admin role is responsible for managing users, projects, and bugs. The Tester role is responsible for reporting bugs and tracking their progress. The Developer role is responsible for resolving bugs and updating their status. The Bug Tracking System is the central component that manages all the data and provides the interface for the users.

Figure 26 : Class diagram

- Use case diagram:

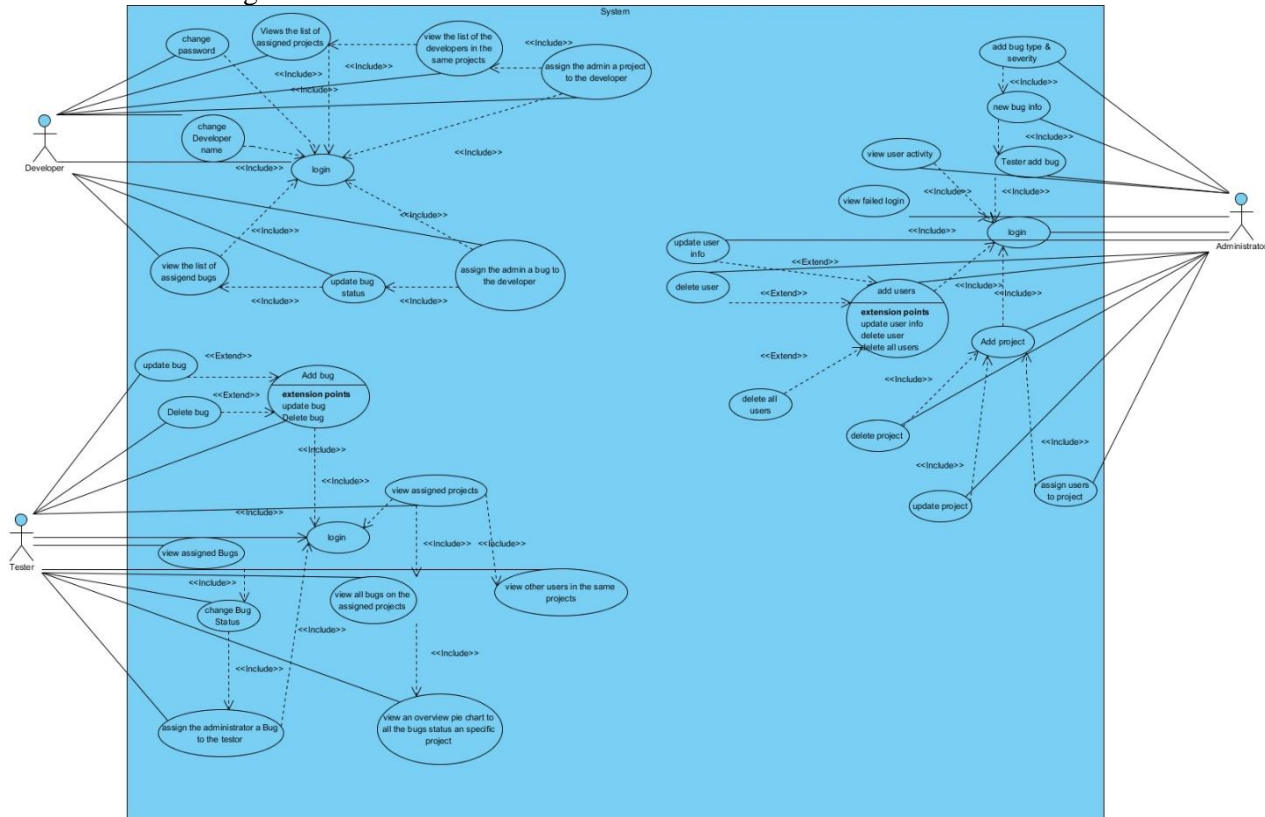


Figure 27 : Use case diagram

11. 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.

12. Future Work:

- Turn the project from a specific use for bugs tracking into general use for project tracking.
- Administrator:
 - Print users data in a table.
 - Print entire project's bug information.
 - Print information about each bug in the project individually.
- Tester:
 - Print information about each bug in the project individually.
 - Enhance the communication between testers and developers by adding comments in each project.

- 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.

13. Time Plan:

13.1 Work Breakdown Structure:

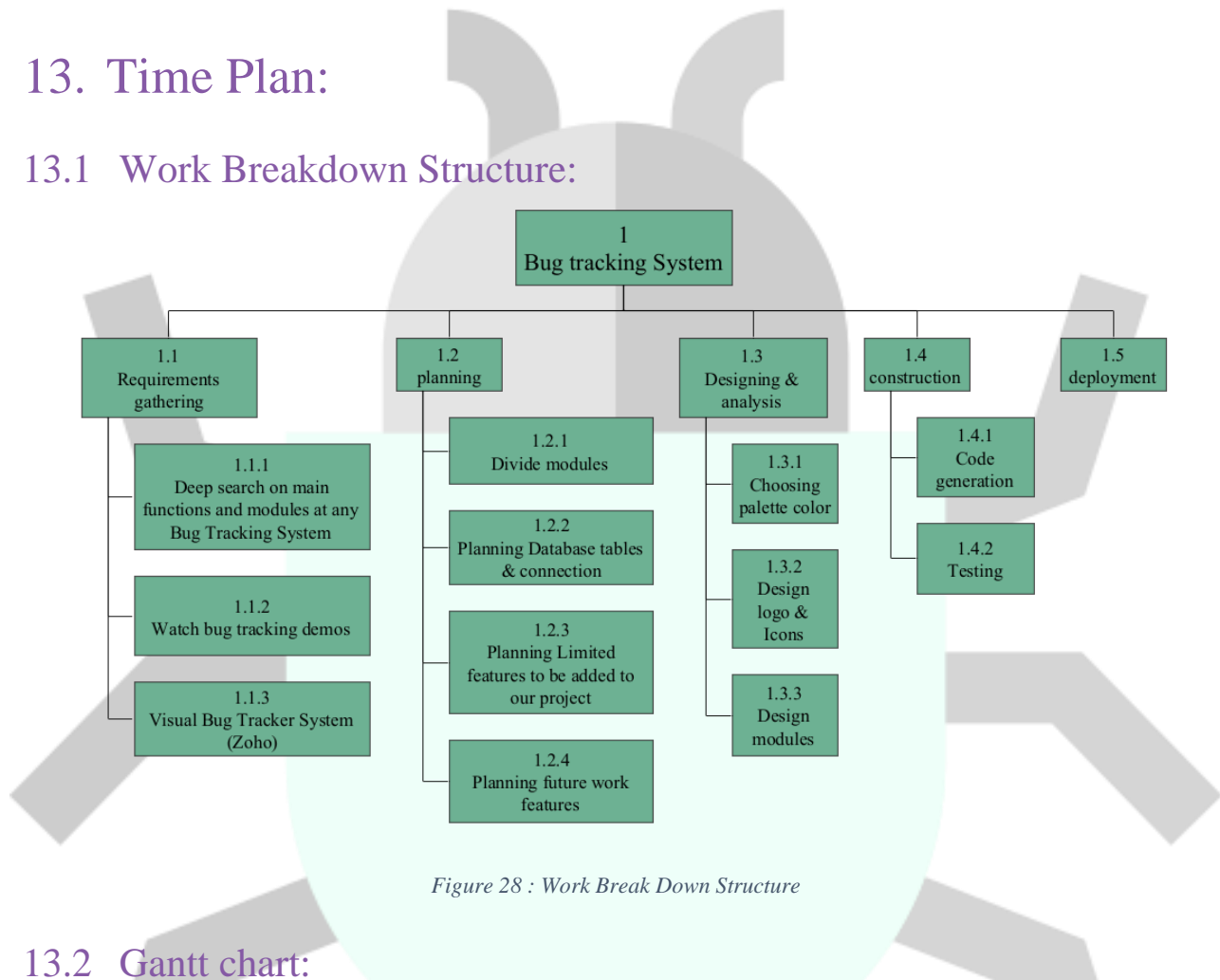


Figure 28 : Work Break Down Structure

13.2 Gantt chart:

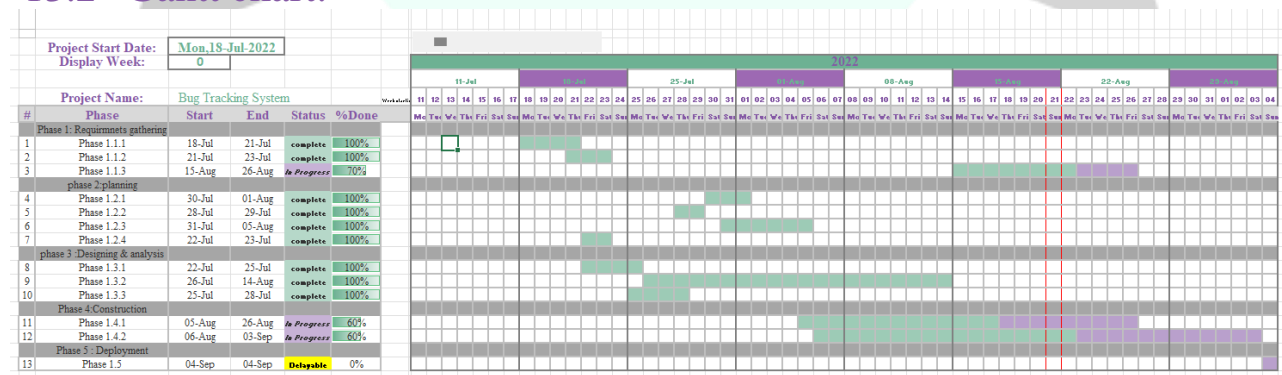


Figure 29 : Gantt chart

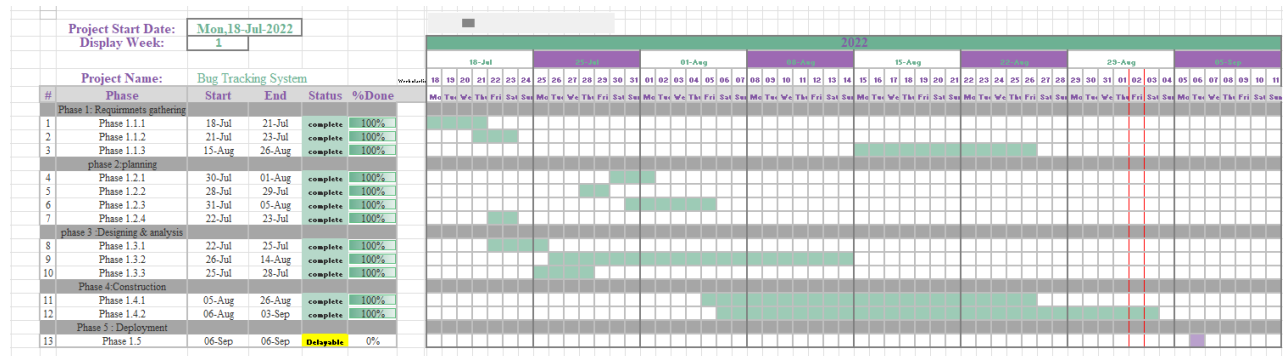


Figure 30 : Updated Gantt Chart

14. References:

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

