**Bug Tracking System Documentation**

**PPT presentation link:**

[**https://docs.google.com/presentation/d/11\_AtOvsMw90BFEzBBgkseH5Z6NOcWPaYYlM7Emup8DY/edit#slide=id.g1f87997393\_0\_848**](https://docs.google.com/presentation/d/11_AtOvsMw90BFEzBBgkseH5Z6NOcWPaYYlM7Emup8DY/edit#slide=id.g1f87997393_0_848)

**1. Introduction**

* **Project Overview**: The Bug Tracking System is designed to streamline the process of identifying, reporting, tracking, and resolving bugs within software projects. The system supports multiple user roles including Project Managers, Developers, and Testers, each with specific tasks and permissions.
* **Objectives**: The primary objective is to create an efficient system for managing bugs across different projects, enabling teams to collaborate effectively and ensure software quality.

**2. System Design**

* **Level 1 System Design Overview**: The system is composed of various modules interconnected to allow seamless flow of information between different roles (User, Project Manager, Developer, Tester). The primary entities are Users, Projects, and Bugs.

**3. User Roles and Permissions**

* **User**:
  + Can register and log in to the system.
  + Depending on the role (Project Manager, Developer, Tester), the user will have access to specific functionalities.
* **Project Manager**:
  + Can create new projects and assign bugs to developers.
  + Has access to view user information, a list of bugs, and project details.
  + Manages the lifecycle of bugs, including assigning and closing bugs.
* **Tester**:
  + Responsible for creating new bugs and assigning them to projects.
  + Can view assigned projects and user information.
* **Developer**:
  + Can view the bugs assigned to them and update their status (e.g., close bugs).
  + Has access to their assigned projects and can view user information.

**4. Module Descriptions**

**4.1 User Module**

* **User Registration**:
  + Allows new users to register into the system with their role defined (Project Manager, Developer, Tester).
* **Login**:
  + Authenticates the user and redirects them to their respective dashboard based on their role.
* **Display User Info**:
  + Users can view their profile information, including their role and the projects they are associated with.

**4.2 Project Manager Module**

* **New Project**:
  + The Project Manager can create new projects, which will be added to the system's database.
* **List of Bugs**:
  + Displays a comprehensive list of all bugs reported within the system.
* **Assign/Close Bug**:
  + The Project Manager can assign bugs to developers and close them once resolved.
* **Display Projects**:
  + Provides a detailed list of all projects managed by the Project Manager.

**4.3 Tester Module**

* **Create Bug**:
  + Testers can report new bugs, linking them to specific projects.
* **Display Assigned Projects**:
  + Allows Testers to view all projects they are assigned to.

**4.4 Developer Module**

* **Display Assigned Projects**:
  + Developers can view the projects assigned to them and focus on resolving bugs.
* **Close Bug**:
  + Once a bug is resolved, the Developer can update its status to 'Closed'.

**4.5 Bug Module**

* **Bug Entity**:
  + Central to the system, it holds all the details of bugs reported, including their status, priority, and history.
* **Bug Lifecycle**:
  + The bug progresses through different states (e.g., New, Assigned, In Progress, Closed) as it is managed by the team.

**5. System Flow**

* **User Interaction Flow**:
  + The flow begins with the user logging into the system. Based on their role, they are presented with different dashboards and functionalities.
* **Bug Tracking Flow**:
  + Testers report bugs, which are then managed by Project Managers and resolved by Developers. The system tracks the entire lifecycle of each bug.

**6. Database Design**

* **Entities and Relationships**:
  + The database comprises tables for Users, Projects, and Bugs, with relationships defined between them.
* **Primary Keys**:
  + Each table has a primary key to uniquely identify records (e.g., UserID, ProjectID, BugID).
* **Foreign Keys**:
  + Foreign keys ensure the integrity of relationships (e.g., ProjectID in the Bug table links to the Projects table).

**7. Implementation Details**

* **Frontend Technologies**:
  + Description of the technologies used to build the user interface (e.g., HTML, CSS, JavaScript).
* **Backend Technologies**:
  + Outline the server-side technology stack (e.g., Java)
* **Database**:
  + Description of the database system used (e.g., MySQL)

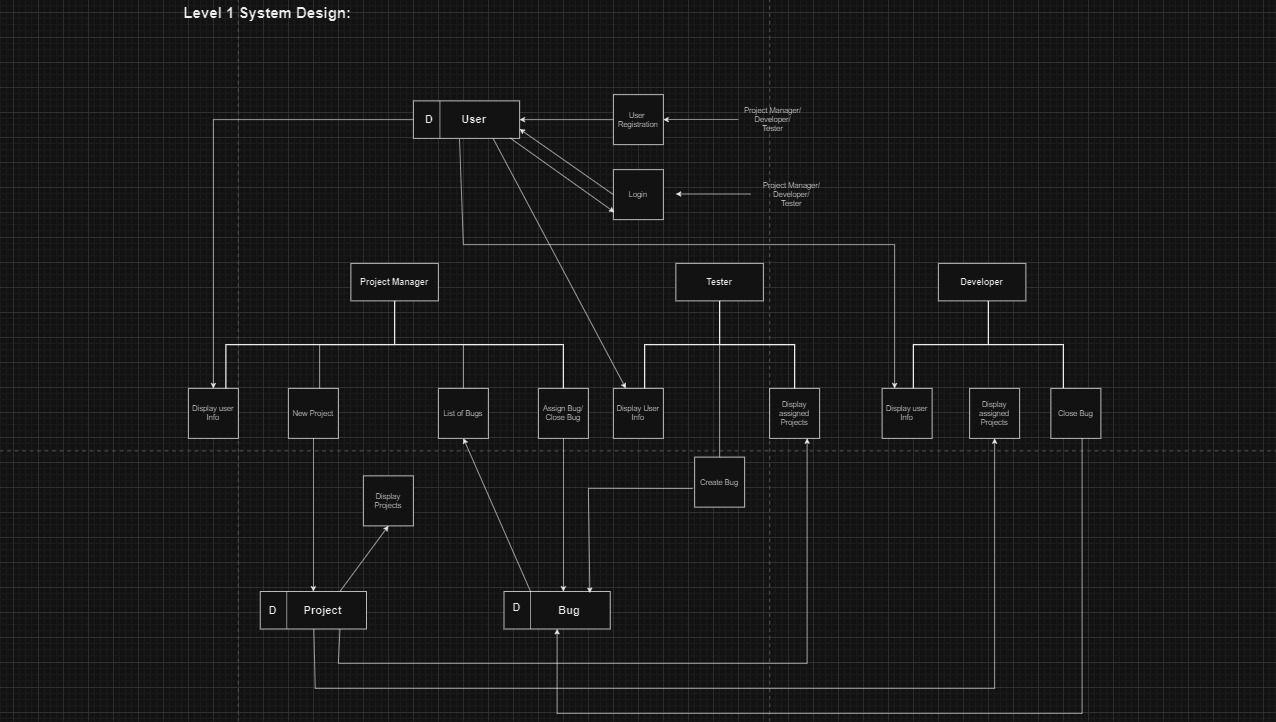
**8. Conclusion**

* **Project Summary**:
  + The Bug Tracking System is an essential tool for managing software quality, ensuring that issues are tracked and resolved efficiently.
* **Future Enhancements**:
  + Consider adding features like automated bug reporting, integration with version control systems, and advanced analytics.

**9. References**

* **Only HSBC training is Enough to implement the Approach.**

**10. Attached System Design**



**USE Case Diagram**

A diagram of a diagram

Description automatically generated

