***ANUDIP FOUNDATION***

A Project Report on

**BUG TRACKING SYSTEM**

By

**Batch:** ANP-D0453

**Student ID:** AF0477036

**Name**: Bhagyashri Ningabasappa Nadageri

**Under the Guidance of**

Mrs. Rajshri Chandrabhan Thete

**BUG TRACKING SYSTEM**

**INTRODUCTION:**

Every software application goes through multiple development and testing phases, and during this process, issues or bugs are bound to appear. A Bug Tracking System (BTS) helps software teams track and manage these bugs efficiently, ensuring they are fixed as quickly as possible.

Our Bug Tracking System, built using Java, MySQL, and Hibernate, acts as a digital assistant for developers and testers. It provides a centralized platform where users can report issues, assign them to the right people, track their progress, and ensure they are resolved before the software is delivered. This system not only saves time but also improves collaboration and software quality.

**ENTITIES:**

* Developer
* Tester
* Project
* Bug
* Report

**ATTRIBUTES OF ENTITIES:**

* **Developer:**

1. dev\_id(Primary Key)
2. first\_name
3. last\_name
4. email
5. username
6. password
7. project\_id(Foreign Key)

* **Tester:**

1. test\_id(Primary Key)
2. first\_name
3. last\_name
4. email
5. username
6. password
7. project\_id(Foreign Key)

* **Project:**

1. project\_id(Primary Key)
2. project\_name

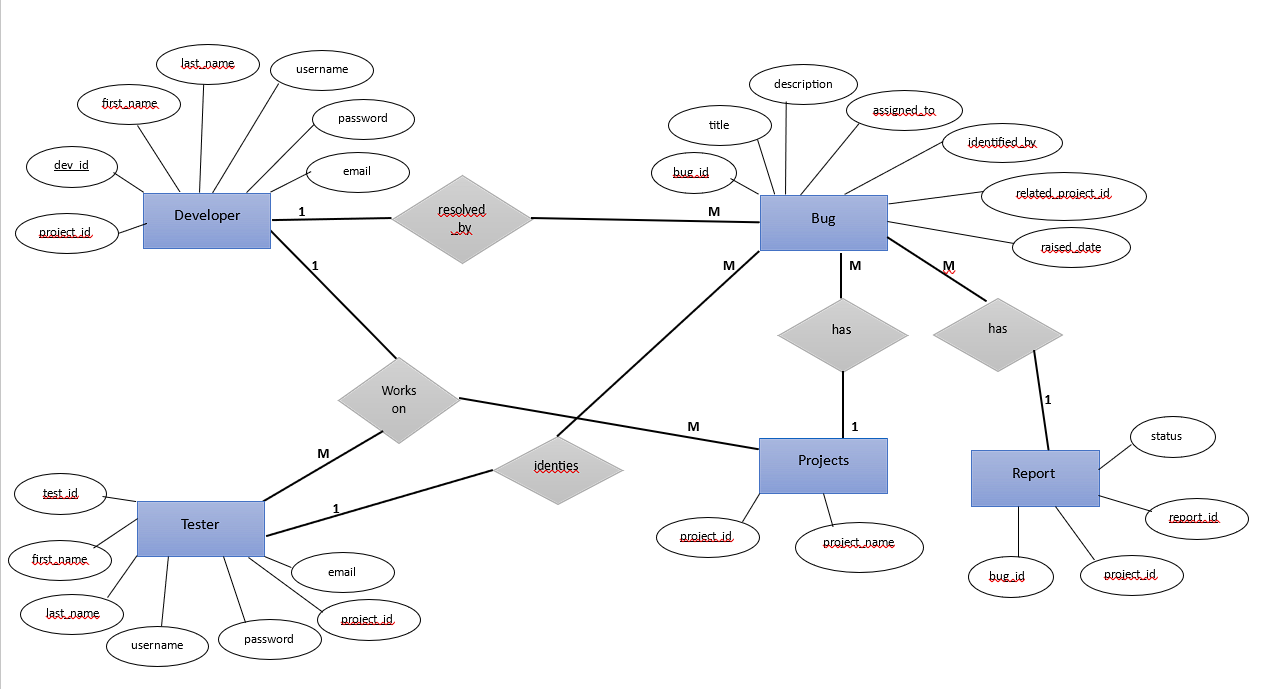
* **Bug:**

1. bug\_id(Primary Key)
2. title
3. description
4. raised\_date
5. dev\_id(Foreign Key)
6. test\_id(Foreign Key)
7. related\_project\_id(Foreign key)

* **Report:**

1. report\_id(Primary Key)
2. project\_id(Foreign Key)
3. bug\_id(Foreign Key)
4. status

**ER DIAGRAM OF BUG TRACKING SYSTEM:**

****

**CONCLUSION:**

Bugs are an inevitable part of software development, but managing them doesn’t have to be a headache. A Bug Tracking System helps development teams stay organized by keeping track of every issue in one place, ensuring nothing slips through the cracks.

With our system, built using Java, MySQL, and Hibernate, teams can easily report, assign, and resolve bugs without relying on endless emails or messy spreadsheets. It streamlines communication, improves collaboration, and helps developers focus on what they do best—building great software.

By making bug tracking simple, efficient, and transparent, this system ensures that software gets delivered with fewer issues and a better user experience. In the end, it’s not just about fixing bugs—it’s about making the entire development process smoother, faster, and more reliable.

**DATABASE CREATION QUERY:**

**mysql> create database bug\_tracking\_system;**

**Query OK, 1 row affected (0.01 sec)**

**mysql> use bug\_tracking\_system;**

**Database changed**

**mysql> CREATE TABLE project(project\_id int NOT NULL primary key,**

**-> project\_name VARCHAR(50));**

**Query OK, 0 rows affected (0.04 sec)**

**mysql> CREATE TABLE developer(dev\_id int NOT NULL Primary Key,**

**-> first\_name VARCHAR(20), last\_name VARCHAR(20),**

**-> email VARCHAR(30),**

**-> username VARCHAR(30), password VARCHAR(30),**

**-> project\_id int,**

**-> FOREIGN KEY(project\_id) REFERENCES project(project\_id));**

**Query OK, 0 rows affected (0.04 sec)**

**mysql> CREATE TABLE tester(test\_id int NOT NULL Primary Key,**

**-> first\_name VARCHAR(20), last\_name VARCHAR(20),**

**-> email VARCHAR(30),**

**-> username VARCHAR(30), password VARCHAR(30),**

**-> project\_id int,**

**-> FOREIGN KEY(project\_id) REFERENCES project(project\_id));**

**Query OK, 0 rows affected (0.05 sec)**

**mysql> CREATE TABLE BUG(bug\_id int NOT NULL Primary key,**

**-> title VARCHAR(30), description TEXT, raised\_date DATE,**

**-> assigned\_to int, identified\_by int, related\_project\_id int,**

**-> FOREIGN KEY(assigned\_to) REFERENCES developer(dev\_id),**

**-> FOREIGN KEY(identified\_by) REFERENCES tester(test\_id),**

**-> FOREIGN KEY(related\_project\_id) REFERENCES project(project\_id));**

**Query OK, 0 rows affected (0.07 sec)**

**mysql> CREATE TABLE report(report\_id int NOT NULL Primary Key,**

**-> project\_id int, bug\_id int, status VARCHAR(20),**

**-> FOREIGN KEY(project\_id) REFERENCES project(project\_id),**

**-> FOREIGN KEY(bug\_id) REFERENCES bug(bug\_id));**

**Query OK, 0 rows affected (0.05 sec)**

**mysql> show tables;**

**+-----------------------------------------+**

**| Tables\_in\_bug\_tracking\_system |**

**+-----------------------------------------+**

**| bug |**

**| developer |**

**| project |**

**| report |**

**| tester |**

**+----------------------------------------+**

**5 rows in set (0.04 sec)**