**Bug Tracking System**

Created by -

Venkatesh M-46004145

INTRODUCTION

**Problem Statement:**

To create a bug tracking system for the software production. The application can be made either as Web Application, Mobile Application or Desktop Application

**Description:**

This project is aimed at developing the bug tracking system .Here an employee raises the ticket and then assignes the resposible person for debugging the code snippets assigned to that particular task .Once the bugs are solved then the employee must be able to close the ticket.

**Required Features:**

* Employee should be able to login through the app.
* System should have details about the project and dashboard showing current bugs.
* Employees should be able to raise ticket for the bug.
* Provide critical levels for bugs (low, medium, high, severe).
* Employee should be able to add note to the ticket.
* Employee should be able to paste code snippets.
* Employee should be able to highlight important words in note using # tag.
* Employee should be able to tag responsible person for bug fix using @ tag.
* When @ is typed then list of employees should be displayed and should be able to select the responsible employee.
* When using # and @ the text color should be changed so it can be highlighted.
* When @ is used and ticket is raised then the responsible employee should be notified.
* Once bug is solved employee should be able to close the ticket.
* Create a good UI for good user experience.
* Plan a good architecture pattern for the project.
* Provide security for project.

**Scope :**

Managers are already present in the database

Managers can add the project

Managers add the developer

**Out of Scope:**

Bugs are considered to be solved only when the compilation is successful

Manager pastes the code snippet to the ticket

Submission is done automatically.

**TechStack Used:**

* Java 8
* Angular 6
* Maven –v3.6.2
* Spring Boot 2.x
* SpringREST
* Spring Data JPA
* Postman
* MySql
* JUnit 5/Mockito
* Sonarqube

**Functional Components Used:**

**Manager :**

Manager can raise the ticket and assign a developer for solving the bug.Manager can paste the code snipets,add note to the ticket Once the bug is solved by the developer then the manager must be able to cancel the ticket.

**Developer** :

Developer gets notified about the ticket raised by the manager.Developer solves the bugs and he must send the code snippet to the manager.

**CLASSES AND METHODS :**

The Bug tracking system consists of following classes and methods

**Ticket Class :**

Attributes :

* ticketId: BigInteger
* ticketNote : String
* assignedTo : Employee //Refer below
* assignedBy : Employee //Refer below
* ticketStatus : String
* codeSnippet : String
* ticketCriticalLevel : CriticalLevel //Refer below
* ticketDeadline : LocalDateTime

**Employee Class** :

Attributes :

* employeeId : BigInteger
* employeeName : String
* employeeRole : String
* loginname : String
* userPassword : String
* emailId : String
* phoneNumber : String

**Manager Class : extends Employee**

Attributes :

* tickets : List<Tickets>
* developers : List<Developer>

**Developer Class : extends Employee**

Attributes :

* ticketAssigned : Ticket
* manager: Manager
* assignStatus : boolean

**CriticalLevel : Enumeration**

Values :

* low
* medium
* high
* severe

**Manager Service Methods**

1. **public** Project addProject(Project project) **throws** Exception: This method is used for adding project to the database.
2. **public** Project endProject(Integer projectId):This method is used for ending the project.
3. **public** Bug addBug(Bug bug,String projectName) **throws** Exception :This method is used for adding the bug to the project.
4. **public** Ticket raiseTicket(Ticket ticket,Integer developerId,Integer bugId) **throws** Exception : This method is used for raising the ticket and assign a bug to the developer.
5. **public** Employee addEmployee(Developer employee) **throws** Exception:This method is used for adding the developer to the database
6. **public** Manager addManager(Manager manager) **throws** Exception : This method is used for adding the manager to the database.
7. **public** List<Project> getProjects(Integer managerId) : This method is used for viewing the projects
8. **public** List<Developer> getDevelopers(Integer managerId)**throws** Exception : This method is used for getting the developers list related to that manager

**Developer Service Methods**

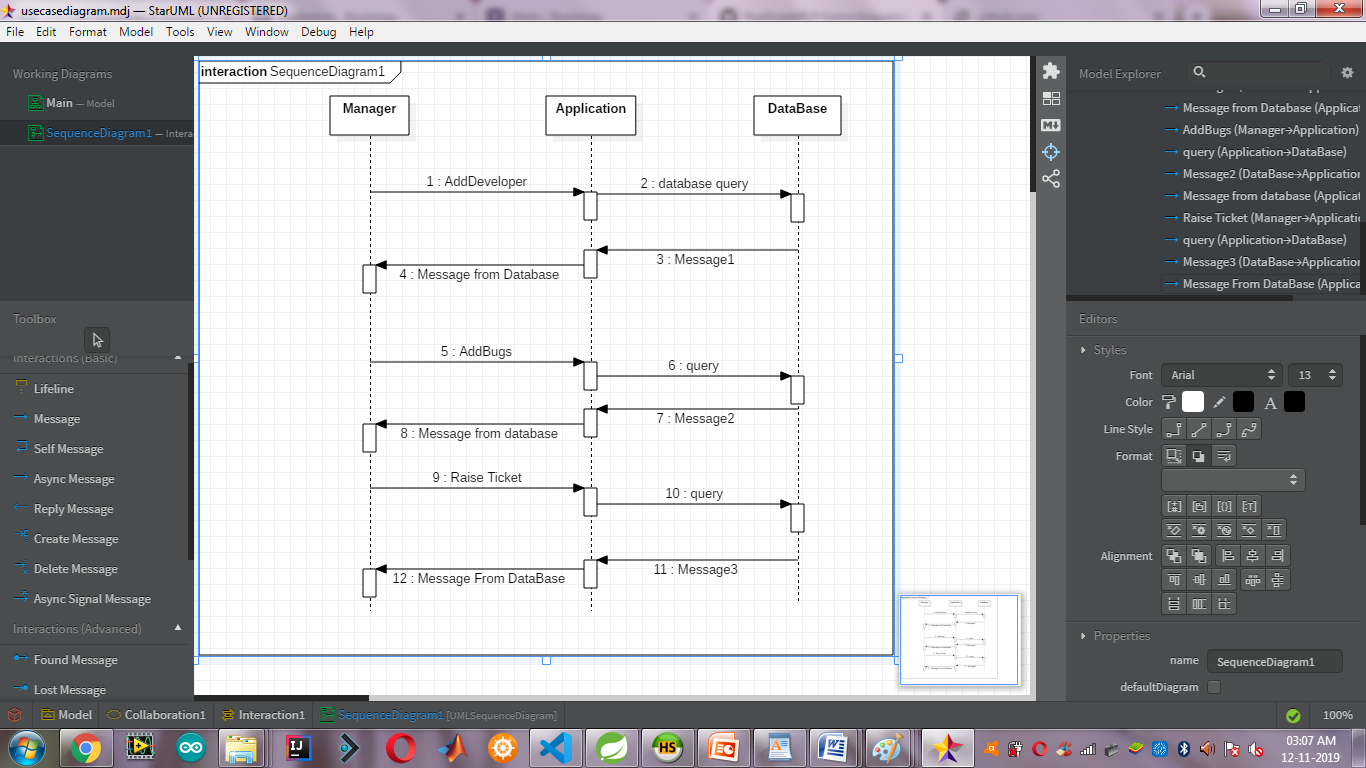
**1.public** **boolean** submit(**int** bugId,**int** developerId,String code : This method is used for submitting the code once compilation is successful.

2. **public** List<String> compile(String code):This method is used for compiling the code

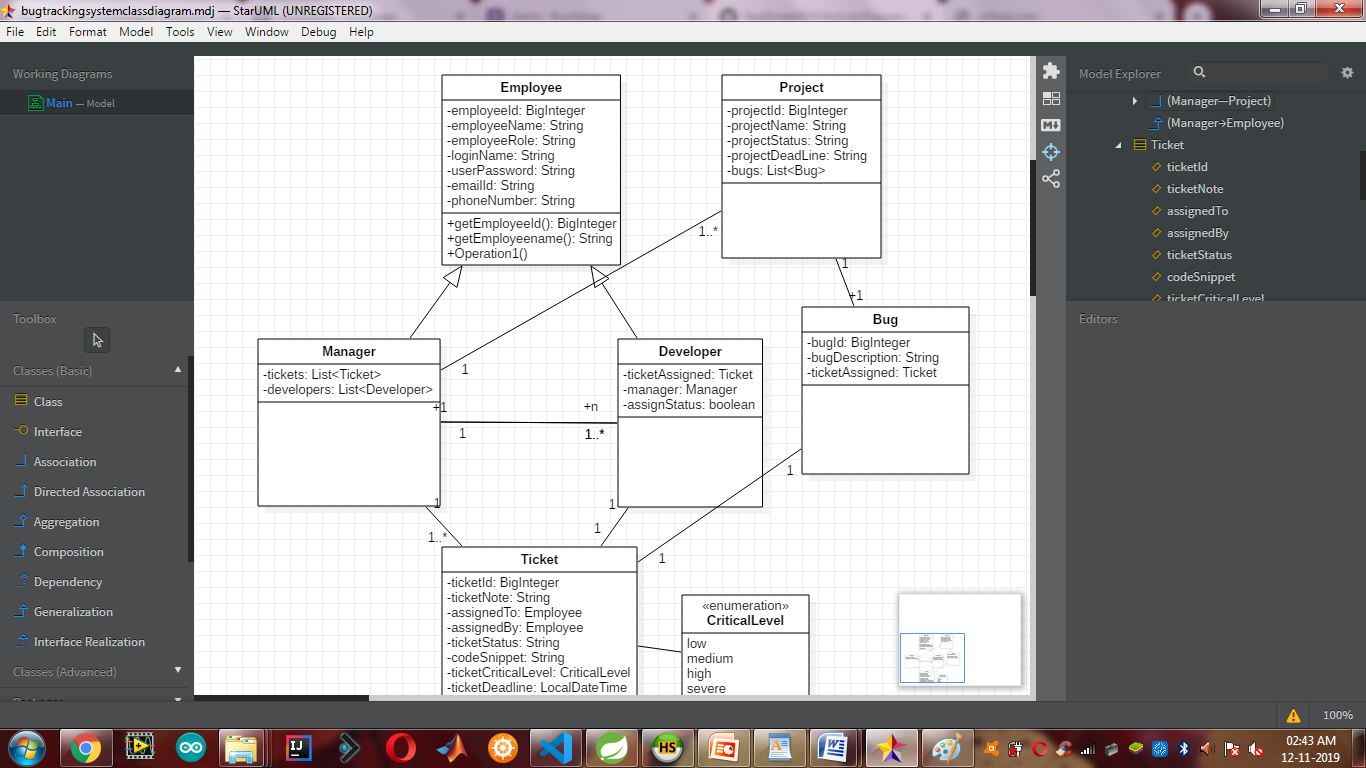
**Validations:**

1. Project end date can not be before the current date
2. Employee name must have atleast 4 charecters
3. Phone Number must have must have 10 digits.
4. EmailId must have @ and .com
5. Bug which is already assigned can not be assigned
6. Ticket can not be raised on a developer who has already been assigned
7. Bug end date can not before than the current date.

**Sequence Diagram:**



**ClassDiagram:**

****

