Bug Tracking System

Project Scope

The project is an online bug tracking system for projects. The application allows software testers to report bugs for a project, project managers to view all bugs, assign bugs to developers and developers to update the bug status.

Description

The online bug tracking system stores all data in a database. The information stored is

Users

The users of the application. The application will have three types of users

1. Project Manager

2. Developers

3. Testers

All users will have a unique user id, name, email, and the type of user.

Projects

The system can have multiple projects. Each project will have a unique project id, project name, description, start date, a status (In-progess or completed), and team members(users).

Teams

Every project has a team of members. The members consist of

1. A single Project Manager

2. Multiple developers

3. A single Tester

Bugs

Bugs or defects are shortcomings or a flaw in the project, which deviates the actual result from the expected one.

Every bug will have a

1. Unique Id

2. Title

3. Description

4. Project name(id)

5. CreatedBy (Testers register the bug)

6. OpenDate

7. AssignedTo (A project manager assigns a bug to a developer)

8. Marked for closing (A developer marks a bug to be closed)

9. ClosedBy (A project manager closes the bug)

10. ClosedOn

11. Status(open/closed)

12. Severity level(critical/major/minor/trivial)

Screens & Use-cases

Home Page

The page should display the application description and have links to

1. Import Users

2. Register

3. Login in.

Import Users

Screen should provide a link called import users once the person clicks on the link request should go to server and in the server it should load the users from the xml file which is available in the server eg. C:\serverlocation\users.xml. you can decide the structure of the xml file.

The imported users should be saved into the database.

Business Rules

1. If a user with the same email address exists, it should not be imported.

2. The email address should be in a proper format.

3. All fields (name, type, email) are mandatory

4. Every user should be assigned a unique id on successful import.

Register

This screen allows a user that has been imported to register with the system and generate a password. The page should show a form capturing the following details

1. Email Address

2. Role (Type)

3. Password

4. Confirm Password

After successful registration he should be delegated to the login screen.

Business Rules

1. All field are mandatory.

2. The email address should exist and match with the role.

3. The user should not have already been registered

4. Strongly typed password

Login

The screen should display a form to login. Fields to display

1. Email

2. Password

After successful login, the user should be delegated to the Bug Tracking system main page.

Business Rules

1. Email and password should match.

2. The user should not be already logged in.

Bug Tracking system Main Page (Role, Project Manager)

 The page should display the user information (Email address, Role and last logged in date and time).

 Should have a link to create a new project.

 Should display a list of all projects managed by the project manager. The project name should be a link which should display details of the project.

Project Details (Role, Project manager)

Displayed when the project manager clicks on the project name in the Bug Tracking system Main Page.

 Part I

o This screen should display the project details (Project Name, start date, project manager and the list of team members with their roles.

 Part II

o It should display a sortable/filterable list of bugs for the projects.

 Part III

o Tasks to perform on bugs.

1. Assign a bug to a developer.

2. Close a bug.

Appropriate UI should be designed for the above tasks with proper validations.

Business Rules

1. Project managers can manage bugs only of projects they manage.

2. Bugs can be assigned only to the developers in the team for that project.

3. A bug can only be closed if it is marked for closing.

New Project (Role, Project Manager)

Displayed when the project manager clicks on the create new project link in the Bug Tracking system Main Page.

The page displays a form to create a new project. Fields to display

1. Project Name.

2. Start Date (Selectable from a dropdown/date/calendar control)

3. Description

4. A UI to assign team members to the project.

Business Rules

1. Start date should be at least 2 days later than the current date.

2. The project status should be set to “In Progress”.

3. Developers can be assigned to only one project.

4. Testers can be assigned to a maximum of 2 projects.

5. Testers can be assigned to projects only under the same project manager.

6. A project manager can manage a maximum of 4 projects.

7. A unique project id should be generated.

Bug Tracking system Main Page (Role, Testers)

The page is displayed when a tester logs in to the system.

 The page should display the user information (Username, Email).

 If the user is not assigned to any project, it should display a message saying so.

 If the user is assigned to a project, it should display the list of projects.

 For each project it should display the list of bugs.

 Link to report a new bug.

Business Rules

1. Only bugs created by the tester can be viewed by him/her.

Report a New Bug(Role, Testers)

The page is displayed when a tester clicks on the Report a New Bug link from the Bug Tracking system Main Page.

The page displays a form to report a new bug. Fields to capture.

1. Project Name

2. Title

3. Description

4. Severity Level

Business Rules

1. Bugs can be reported only by the tester

2. The tester can report bugs only for projects he/she is assigned to.

3. The bugs can be reported only for projects with status in-progress

4. A unique id should be generated

5. The createdBy should the tester

6. The createdOn should be current date and time.

Bug Tracking system Main Page (Role, Developers)

The page is displayed when the developer logs in to the system.

 The page should display the user information (Username, Email).

 If the user is not assigned to any project, it should display a message saying so.

 If the user is assigned to a project, it should display the project details

o Project Name, Manager, Start Date, List of members

 Tasks

o The developer should be able to mark a bug for closing.

o A proper UI with validations should be provided.

Business Rules

1. Developer can only view bugs assigned to him.

2. He/She can mark a bug for closing .

Test Cases

1. Register

a. All field are mandatory.

b. The email address should exist and match with the role.

c. The user should not have already been registered

2. Create New Project

a. Start date should be at least 2 days later than the current date.

b. The project status should be set to “in-progress”.

c. Developers can be assigned to only one project.

d. Testers can be assigned to a maximum of 2 projects.

e. Testers can be assigned to projects only under the same project manager.

Application Design

1. Use an RDBMS server to store all business data viz. Users, Projects, Teams etc. Data should be normalized. Decide the database design (tables and columns) based on the application and business requirement.

2. The application should be object oriented and easily extensible to add new features like new roles and other use cases

3. The web application should be based on MVC design pattern

4. Business rules should be implemented in a separate layer.

5. Implement a proper data-access layer.

6. Use proper exception handling.

7. Provide a logout option

8. Any one page should be designed as responsive

Functional Requirement

1. Security

a. Unauthorized users should not have to access the system

2. Logging

a. All unhandled exceptions should be logged.

3. Transactions

Guidelines

1. Use layered architecture with loose coupling.

2. Functional requirements should be ideally implemented using Aspect-Oriented Programming.

3. Validate all user inputs with proper error handling.

4. The UI should use a layout with a header, footer and sidebar (with navigation links), which should be maintained on all pages.

5. Optional: Explore the various CSS frameworks for designing the UI, The UI should look elegant.

6. Optional: Transactions should be declarative transactions.

Code Fury instructions/Guidelines.

• Create normalized tables and required integrity constraints where ever required

• Submit the table design in the form of a diagram showing the relations ships in a word document/PPT slide.

• Show the relationship between important classes in the form of a diagram in a word /PPT

• Show the layers and the important classes and interfaces for integration between the layers.

• You can skip the implementation of the use cases which are marked red.

• Pay attention to programming standards while coding. Never compromise.

• Plan your time properly for the design, coding and testing phases. Remember, good design leads to good quality code which leads to stable and robust code.

• Show the flow in the form of a diagrams.

• You can make necessary assumptions where ever required ( i.e. if the data is not available you can pre populate the table with some data and if the use case is not implemented by you, assume that it is implemented by another team and it is working fine and the data is available in the data base)

• Any queries you can ask or mail or Ping Pavan Kumar P Nuguru (SME - 9848275467) /Sameer T Ratnaparkhi (Pune grads contact point: 9850914217). All mails should be send from HSBC laptops only.

• Do not try to send data into HSBC network from vendor /personal laptops or vice-versa under any circumstances. Build your project on vendor laptops (all external location) and and HSBC laptops (all Internal batches)

• Start working on the project immediately

• Projects are evaluated based on the approach, effort, design, loosely coupled, Naming conventions, comments, flow, layers etc. (but not only on completion)

• Develop any one screen as a responsive screen.

Ensure that each person in the team has contributed to the coding part. No team member is allowed to do only documentation or non-technical part.