

|  |
| --- |
| **Name:** Bug\_Tracking\_System  **Description:** Generic parent type which represents any possible state of the system. It contains Reporting\_Subsystem, Bug\_Subsystem, Search\_Subsystem and Profile\_Subsystem. Changes made to the Bug\_Tracking\_System will also alter the state of each subsystem. Any changes or updates made to each subsystem will be visible to the user. |
| **Name:** Seach\_Subsystem  **Description:** Subsystem of Bug\_Tracking\_System. Allows the user to query all available Bug\_Reports and relevant attributes. Each Bug\_Report is indexed with a postID, which can be retrieved. |
| **Name:** Reporting\_Subsystem  **Description:** Subsystem of Bug\_Tracking\_System. Allows the user to generate a real time report of the state of the current system. It will contain attributes: indexing posts, status of bugs, number of users etc, with methods responsible for calculating and generating graphs/analytics. |
| **Name:** Bug\_Subsystem  **Description:** Subsystem of Bug\_Tracking\_System. Allows the user to manage Bug\_Report in a number of ways; these methods are: Create new bug report, delete bug report, mark bug report as completed, edit bug report. It also includes attributes about corresponding Bug\_Report ID’s. The Bug\_Subsystem is made up from an array of Bug\_Report. |
| **Name:** Bug\_Report  **Description**: Contains all data required to display the Bug\_Report to the user through the Bug\_Subsystem, this includes attributes such as Post ID, Title, Contents, Poster, Timestamp.  Bug\_Report is also parent of classes Comment, Interest and Apply\_Patch. |
| **Name:** Comment  **Description:** Allows the user to post a comment on an already existing Bug\_Report. When the Bug\_Report is viewed, all associated comments will also be view. This class will contain attributes Comment, Poster and Timestamp. The user can also use methods Edit Comment and Delete Comment. |
| **Name:** Interest  **Description:** Allows the user to register interest on any already existing Bug\_Report instances. It stores attributes user, time of interest, qualifications and questions. A submission can be made using the submit method and interest can also be withdrawn using the withdraw method. |
| **Name:** Apply\_Patch  **Description:** Allows the user to upload any final comments about the bug before closing the bug. This includes changes and issues fixed. Like Interest it contains attributes such as note, user, time of patch. A Bug\_Report can be closed using the closeBug method. |
| **Name:** Profile\_Subsystem  **Description**: Subsytem of Bug\_Tracking\_System. Allows the user to manage user, view users, register and login depending on their permissions. The system keeps attributes of Total Users, Registered Users, Permissions and Backup. A number of methods allows the user to complete the functions mentioned above. |
| **Name:** User  **Description:** A superclass of Registered and Unregistered\_User. A number of Permission methods occupy this class to determine if current user is Registered or Unregistered. |
| **Name**: Unregistered\_User  **Description:** Subclass of User. An unregistered user could be considered to be in *View only* mode, and hence has not attributes. This class will check permissions on any action and also have a method which redirects the user to the Profile\_Subsystem if they wish to sign up. |
| **Name:** Registered User  **Description:** Subclass of User. A registered user is a superclass of the 4 User types, Bug\_Reporter, Reviewer, Triagers and Developers. This class includes all attributes about the user including username, password, user\_id, post\_counts, email, any other profile information, bugs\_worked\_on etc. With methods to allow the user to change personal details, including password; there are also methods which are responsible for the base functionality, taking into account their account type and permissions. |
| **Name:** Bug\_Reporter, Reviewer, Triagers, Developers  **Description:** All subclasses of superclass Registered user. All attributes and methods are inherited from the super class, with additional permission attributes and methods depending on the user type. |
| **Name:** User\_Permissions  **Description:** A dependency on class User, this class lists attributes of all user permissions as well as bool flags to determine what permissions the current instance of User has. This class has methods to prevent all users from posting/making changes and interacting with the server (To be determined and designed at a later date). |