Section 1

Design and Implementation

for

Bug Tracker Application

Version 1.0 approved

Prepared by Group 12

CSCI222, University of Wollongong

27 October 2016

Table of Contents

[Table of Figures 3](#_Toc465367367)

[1. Architectural Design 4](#_Toc465367368)

[1.1 Description 4](#_Toc465367369)

[1.2 Architectural Design Diagram 5](#_Toc465367370)

[2. Use Cases 6](#_Toc465367371)

[2.1 User Profile System 6](#_Toc465367372)

[2.2 Search system 9](#_Toc465367373)

[2.3 Bug System 10](#_Toc465367374)

[2.4 Use case Diagram 13](#_Toc465367375)

[3. Sequence Diagram 14](#_Toc465367376)

[3.1 Add Bug 14](#_Toc465367377)

[3.2 Edit Profile 14](#_Toc465367378)

[3.3 Login 15](#_Toc465367379)

[3.4 Register 15](#_Toc465367380)

[3.5 View Profile 16](#_Toc465367381)

[3.6 View Bug 16](#_Toc465367382)

[3.7 Search Bug 17](#_Toc465367383)

[3.8 Search User 17](#_Toc465367384)

[4. Class Diagram 18](#_Toc465367385)

[5. State Diagrams 21](#_Toc465367386)

[5.1 Bug State 21](#_Toc465367387)

[5.2 User State 22](#_Toc465367388)

[6. Data Persistence 23](#_Toc465367389)

[6.1 Entity Relationship Diagram 23](#_Toc465367390)

[6.2 Data Dictionary 24](#_Toc465367391)

[7. User Interface Design 29](#_Toc465367392)

[7.1 Login Interface 29](#_Toc465367393)

[7.2 Main Interface / Search Bug Interface 30](#_Toc465367394)

[7.3 Add Bug Interface 30](#_Toc465367395)

[7.4 Profile Interface 31](#_Toc465367396)

[7.5 Search User Interface 31](#_Toc465367397)

[7.6 Bug Interface 32](#_Toc465367398)

[7.7 Report Interface 32](#_Toc465367399)

# Table of Figures

[Figure 1 Architectural Design 5](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367337)

[Figure 2 Use Case Diagram 13](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367338)

[Figure 3 Add Bug Sequence Diagram 14](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367339)

[Figure 4 Edit Profile Sequence Diagram 14](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367340)

[Figure 5 Login Sequence Diagram 15](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367341)

[Figure 6 Register Sequence Diagram 15](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367342)

[Figure 7 View Profile Sequence Diagram 16](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367343)

[Figure 8 View Bug Sequence Diagram 16](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367344)

[Figure 9 Search Bug Sequence Diagram 17](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367345)

[Figure 10 Search User Sequence Diagram 17](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367346)

[Figure 11 Class Diagram Part 1 18](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367347)

[Figure 12 Class Diagram Part 2 19](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367348)

[Figure 13 Class Diagram Part 3 20](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367349)

[Figure 14 Bug State Diagram 21](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367350)

[Figure 15 Bug State Diagram 21](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367351)

[Figure 16 User State Diagram 22](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367352)

[Figure 17 User State Diagram 22](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367353)

[Figure 18 Entity Relationship Diagram 23](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367354)

[Figure 19 Data Dictionary Part 1 24](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367355)

[Figure 20 Data Dictionary Part 2 25](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367356)

[Figure 21 Data Dictionary Part 3 26](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367357)

[Figure 22 Data Dictionary Part 4 27](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367358)

[Figure 23 Data Dictionary Part 5 28](#_Toc465367359)

[Figure 24 Login Interface Design 29](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367360)

[Figure 25 Main Window interface Design 30](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367361)

[Figure 26 Add Bug Interface Design 30](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367362)

[Figure 27 Profile Interface Design 31](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367363)

[Figure 28 Search User Interface Design 31](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367364)

[Figure 29 Bug Interface Deign 32](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367365)

[Figure 30 Report Interface Design 32](file:///C:\Users\njs10\Dropbox\UNI\2016\Session%202\CSCI222\A2\Design%20and%20Implementation.docx#_Toc465367366)

# Architectural Design

## Description

The architecture used to create Bug Tracker is based on the B-C-E Model View Controller Design pattern.

Boundary classes are all the graphical user interfaces incorporated into the software, each boundary class identifies a use case. As the application is database driven, with each interface continually drawing and sending data, just one database controller class is used to handle this functionality.

The database controller class interacts with the boundary classes to provide access to the database. This controller class also interacts with the User and Bug entity classes, creating and returning new bug and user objects. The boundary and entity classes do not interact at all, as the main business logic is contained in the controller class. Business logic for this application is predominantly SQL statements with minor use case specific functionality such as checking password length, storing reputation points, etc. This is why only one controller class is used to keep the design simple.

The above diagram displays the architecture from top down with the log-in or Register screen being the point of entry. A user can also enter as a guest to view the main window which contains the search bug facility. Guest users can view the main window, search bug and search user interfaces.

Once a user is logged in they gain access to sub-systems via a menu on the main window. The user object created on log-in is passed to each sub-system from the main window and checked for access rights against its type. E.g. Reporter menu will only display sub-system items a reporter can use.

In summary, user interaction is via the boundary classes in the form of a graphical user interface. Boundary classes interact with the controller to request and accept data as the controller contains the main business logic. Entity classes contain all functionality for the creation of a User or Bug, e.g. setters and getter and interact only with the controller class.

## Architectural Design Diagram

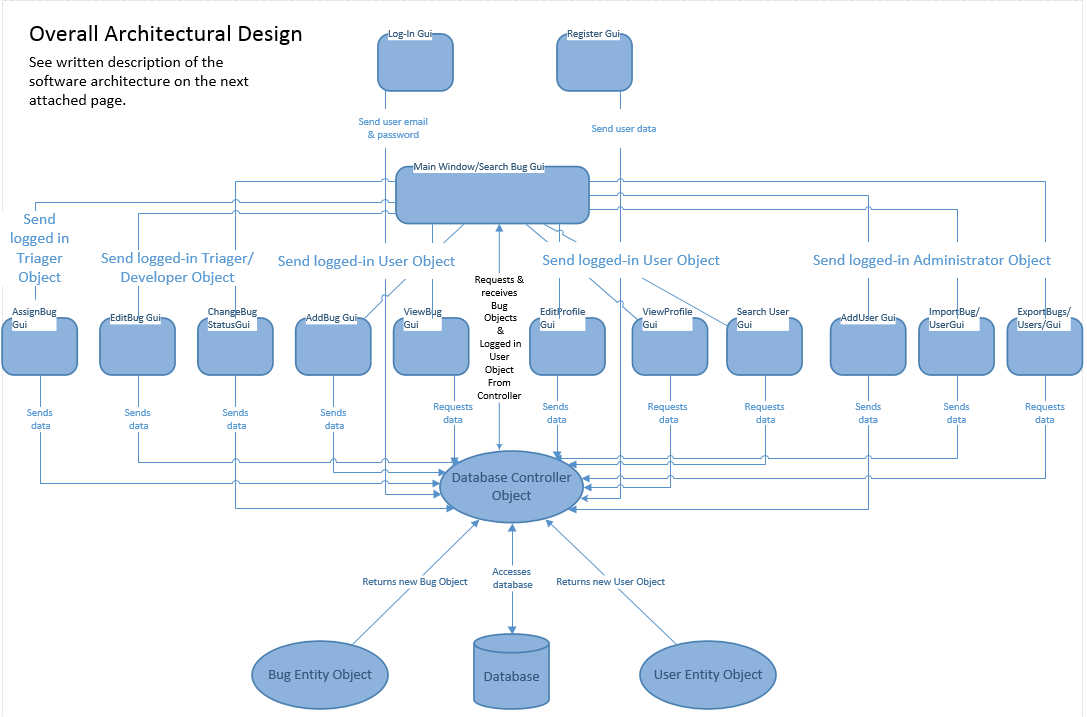


Figure 1 Architectural Design

# Use Cases

## User Profile System

|  |  |
| --- | --- |
| **Name:** Login | **ID: 1** |
| **Stakeholders and goals:** Guest - Can login to the system | |
| **Description:** A user wants to login to the system must be with valid email address and password. | |
| **Actors:** Guest | |
| **Trigger:** User open the program | |
| **Normal flow:**   1. User inputs password and email 2. System check password and email 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows: if the password goes wrong, it will go to invalid password** | |

|  |  |
| --- | --- |
| **Name:** Register | **ID: 2** |
| **Stakeholders and goals:** Guest - register to the system | |
| **Description:** A guest wants to register to the system must be with valid email address and password. | |
| **Actors:** Guest | |
| **Trigger:** Guest open the register interface | |
| **Normal flow:**   1. Guest inputs password and email 2. System check password and email 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** If user do not enter a password, it will go to invalid password.  If user enter an email already in database, it will go to bad email. | |

|  |  |
| --- | --- |
| **Name:** Modify Profile | **ID: 3** |
| **Stakeholders and goals:** User – modify profile to the system | |
| **Description:** A user wants to update personal information to the system | |
| **Actors:** User | |
| **Trigger:** User open the profile interface. | |
| **Normal flow:**   1. User inputs personal information 2. System saves into database 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Edit | **ID: 4** |
| **Stakeholders and goals:** User - can edit personal information that saved before | |
| **Description:** A user wants to change personal information inputted before | |
| **Actors:** Users | |
| **Trigger:** User press button | |
| **Normal flow:**   1. User inputs personal information 2. System Re-writes into database 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Change password | **ID: 5** |
| **Stakeholders and goals:** User – Can change own password | |
| **Description:** A user wants to change their password | |
| **Actors:** Users | |
| **Trigger:** User clicks the change password button | |
| **Normal flow:**   1. User inputs old password 2. User inputs new password 3. User inputs new password again 4. System retrieves input and validates password 5. System updates user password 6. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Enable/Disable User | **ID: 6** |
| **Stakeholders and goals:** system administer - can enable or disable user | |
| **Description:** system administer wants to enable or disable a user | |
| **Actors:** system administer | |
| **Trigger:** system administer go to the enable/disable interface | |
| **Normal flow:**   1. system administer chose who to enable or disable 2. System change in database 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Add new user | **ID: 7** |
| **Stakeholders and goals:** system administer - can add new user | |
| **Description:** system administer wants to add a user | |
| **Actors:** system administer | |
| **Trigger:** system administer go to the add new interface | |
| **Normal flow:**   1. system administer all necessary information for a user 2. System check in database to avoid error 3. System save in database 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Import user | **ID: 8** |
| **Stakeholders and goals:** system administer - can import user | |
| **Description:** system administer wants to import a user | |
| **Actors:** system administer | |
| **Trigger:** system administer go to the import interface | |
| **Normal flow:**   1. system administer chose where to import user 2. System check in database to avoid error 3. System saves in database 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Remove Existing User | **ID: 9** |
| **Stakeholders and goals:** system administer – can remove existing user | |
| **Description:** system administer wants to remove existing user | |
| **Actors:** system administer | |
| **Trigger:** system administer go to the remove interface | |
| **Normal flow:**   1. system administer chose who to remove 2. System check in database to avoid error 3. System changes in database 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Search system

|  |  |
| --- | --- |
| **Name:** Search Bug | **ID: 10** |
| **Stakeholders and goals:** User and guest - can search bug from the system | |
| **Description:** A user or guest wants to search bugs from the system by type, bug number or keywords and search subsystem will return result let user or guest filter bugs | |
| **Actors:** Guest and Users | |
| **Trigger:** Guest or users enter information into search bar | |
| **Normal flow:**   1. User or Guest inputs input into search bar and chose search type 2. System search information in database 3. System gives result to searcher 4. User or guest browse results 5. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** If any results return, it will go to filter search. | |

|  |  |
| --- | --- |
| **Name:** Search user profile | **ID: 11** |
| **Stakeholders and goals:** User - can search other users’ profile | |
| **Description:** A user wants to search other user’s profile | |
| **Actors:** Users | |
| **Trigger:** User input to user search bar | |
| **Normal flow:**   1. User input information to search 2. System check information in database 3. System return results 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Bug System

|  |  |
| --- | --- |
| **Name:** Report bug | **ID: 12** |
| **Stakeholders and goals:** User – can Report a bug | |
| **Description:** User wants to Report a bug | |
| **Actors:** User | |
| **Trigger:** User find a bug and go to report interface | |
| **Normal flow:**   1. User gives information of bug 2. System save all information 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Comment bug | **ID: 13** |
| **Stakeholders and goals:** User – can Comment a bug | |
| **Description:** User wants to Comment a bug | |
| **Actors:** User | |
| **Trigger:** User have comment on a bug | |
| **Normal flow:**   1. User write comment 2. System save comment 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Filter bug | **ID: 14** |
| **Stakeholders and goals:** User or Guest – can Filter a bug | |
| **Description:** User or Guest wants to Filter a bug | |
| **Actors:** User and Guest | |
| **Trigger:** User or Guest want to have a look on bug | |
| **Normal flow:**   1. User or guest select a bug 2. User or guest see the detail of the bug 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Upload patch | **ID: 15** |
| **Stakeholders and goals:** Developer can upload a patch | |
| **Description:** Developer wants to upload a patch | |
| **Actors:** Developer | |
| **Trigger:** Developer have patch to upload | |
| **Normal flow:**   1. Developer select patch 2. System upload the patch to database 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Verify patch | **ID: 16** |
| **Stakeholders and goals:** Reviewer can verify a patch | |
| **Description:** Reviewer wants to verify a patch | |
| **Actors:** Reviewer | |
| **Trigger:** Reviewer have patch to verify | |
| **Normal flow:**   1. Reviewer select a bug 2. System organize the patch of the bug form database 3. Reviewer verify the patch 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Verify bug | **ID: 17** |
| **Stakeholders and goals:** Triager can verify a bug | |
| **Description:** Triager wants to verify a bug | |
| **Actors:** Triager | |
| **Trigger:** Triager have bug to verify | |
| **Normal flow:**   1. Triager get a bug 2. Triager verify the bug 3. System saves the information 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Update Existing Bug | **ID: 18** |
| **Stakeholders and goals:** Triager can update existing bug | |
| **Description:** Triager wants to update existing bug | |
| **Actors:** Triager | |
| **Trigger:** Triager go to update bug interface | |
| **Normal flow:**   1. Triager select a bug 2. Triager change the bug 3. System saves the information 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Assign Bug to Developer | **ID: 19** |
| **Stakeholders and goals:** Triager can give assignment to developer | |
| **Description:** Triager wants to give assignment to developer | |
| **Actors:** Triager | |
| **Trigger:** Triager go to update assignment interface | |
| **Normal flow:**   1. Triager select a bug 2. Triager select a developer 3. System mark the developer and bug 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** Subscribe bug | **ID: 20** |
| **Stakeholders and goals:** User – can subscribe a bug | |
| **Description:** User wants to subscribe a bug | |
| **Actors:** User | |
| **Trigger:** User browse a bug | |
| **Normal flow:**   1. User press the subscribe button 2. System mark the bug 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **Name:** View bug | **ID: 21** |
| **Stakeholders and goals:** User – can View a bug | |
| **Description:** User wants to View a bug | |
| **Actors:** User | |
| **Trigger:** User browse a bug | |
| **Normal flow:**   1. User press the view button 2. System gather bug data and displays it 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use case Diagram

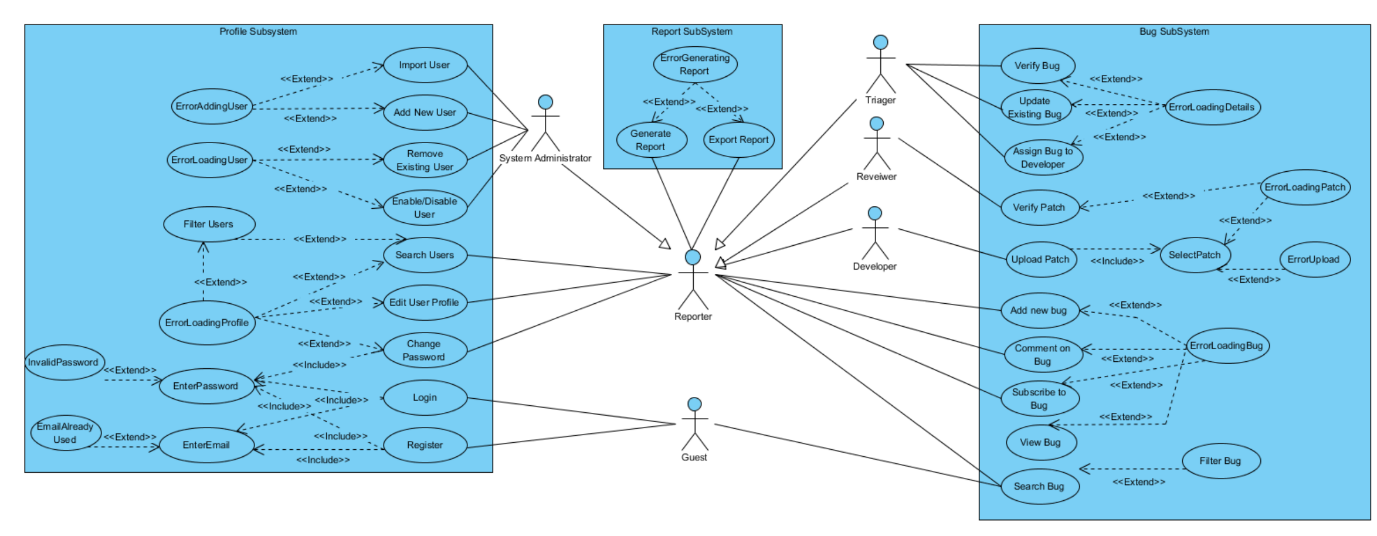


Figure 2 Use Case Diagram

# Sequence Diagram

## C:\Users\njs10\Documents\GitHub\BugTracker\Diagrams\AddBugSequenceDiagram.PNGAdd Bug

Figure 3 Add Bug Sequence Diagram

## C:\Users\njs10\Documents\GitHub\BugTracker\Diagrams\Edit profile.pngEdit Profile

Figure 4 Edit Profile Sequence Diagram

## C:\Users\njs10\Documents\GitHub\BugTracker\Diagrams\Login.pngLogin

Figure 5 Login Sequence Diagram

## C:\Users\njs10\Documents\GitHub\BugTracker\Diagrams\Regisiter.pngRegister

Figure 6 Register Sequence Diagram

## C:\Users\njs10\Documents\GitHub\BugTracker\Diagrams\View profile.pngView Profile

Figure 7 View Profile Sequence Diagram

## C:\Users\njs10\Documents\GitHub\BugTracker\Diagrams\ViewBugSequence.PNGView Bug

Figure 8 View Bug Sequence Diagram

## Search Bug

Figure 9 Search Bug Sequence Diagram

## Search User

Figure 10 Search User Sequence Diagram

# C:\Users\njs10\Documents\GitHub\BugTracker\Documents\Completed%20Documents\bugTrackerClassDiagram_latest.pngClass Diagram

Figure 11 Class Diagram Part 1

Database Class  
See Next Page

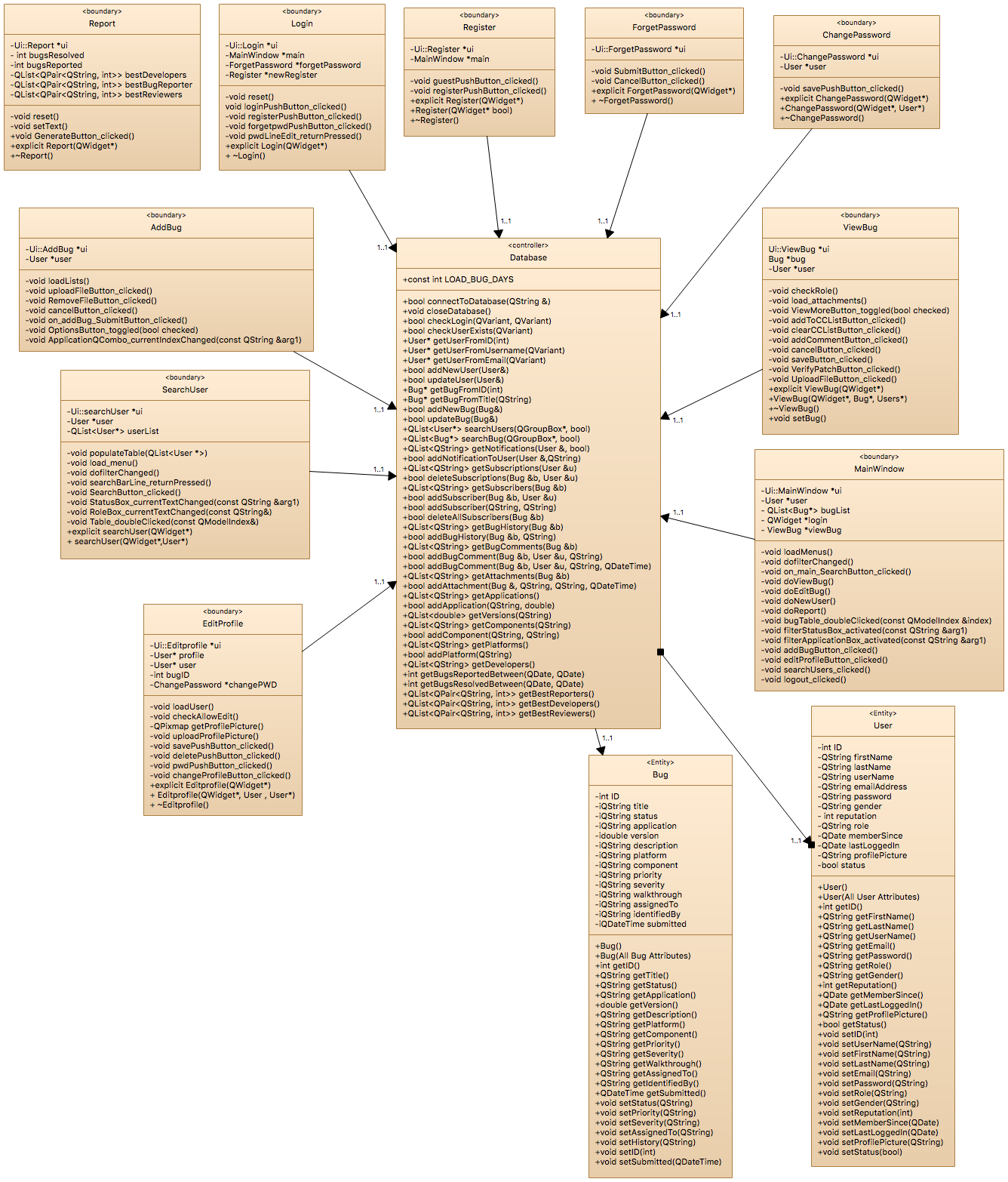


Figure 12 Class Diagram Part 2

See Next Page

See Previous Page

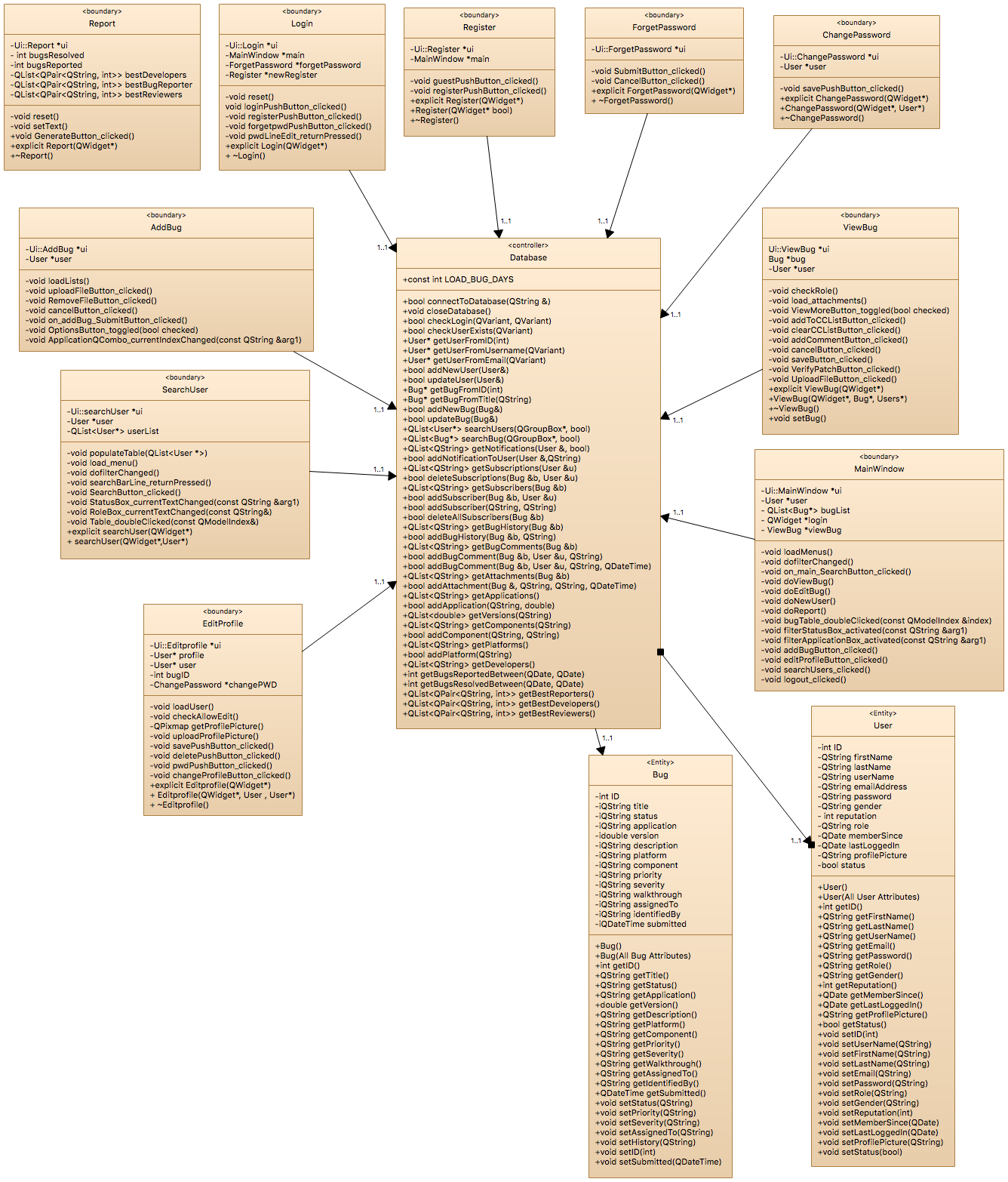


Figure 13 Class Diagram Part 3

See Previous Page

# State Diagrams

## Bug State

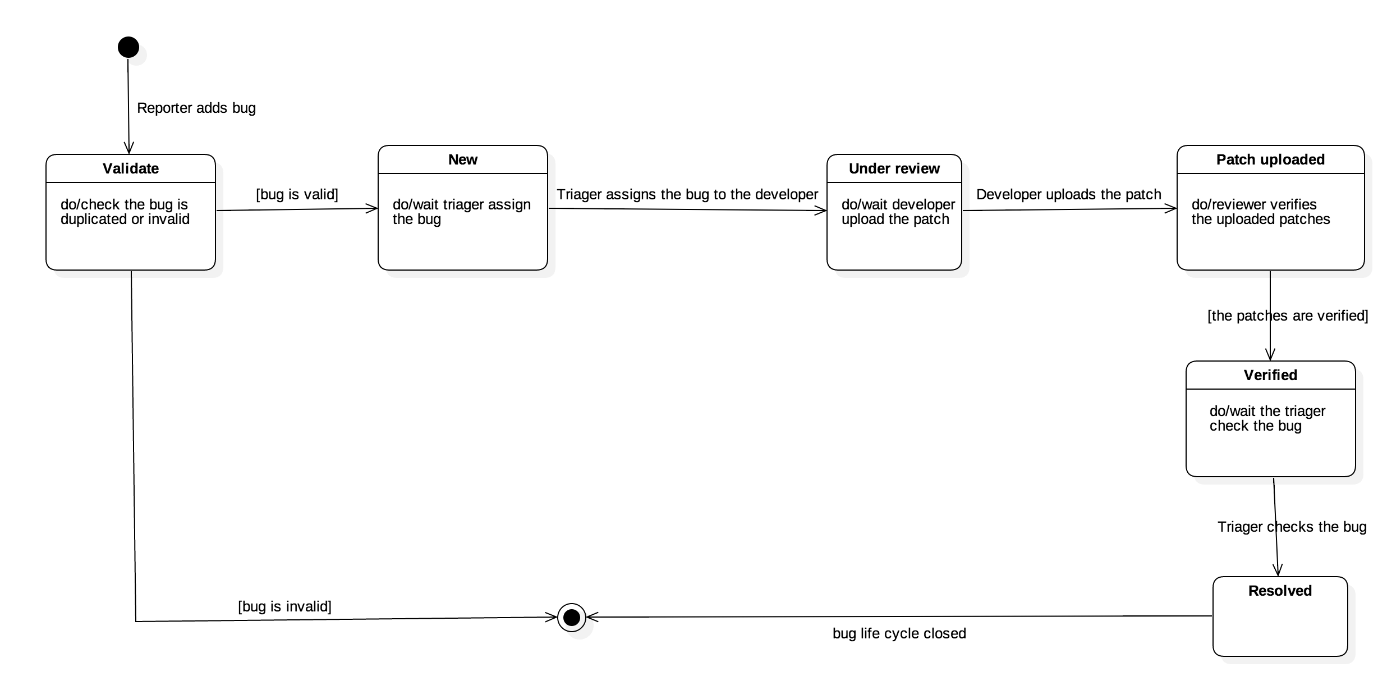


Figure 14 Bug State Diagram

Figure 15 Bug State Diagram

## User State

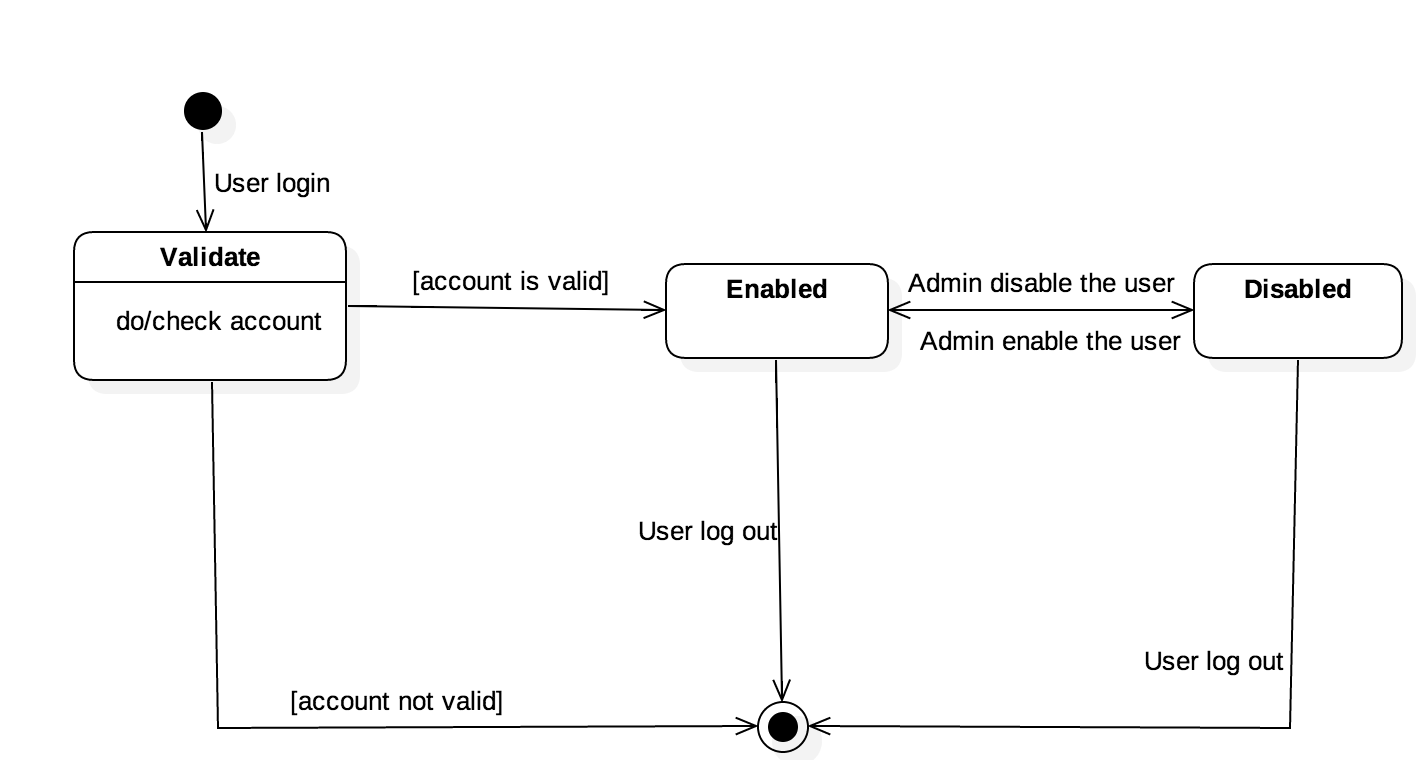


Figure 16 User State Diagram

Figure 17 User State Diagram

# Data Persistence

## Entity Relationship Diagram

Figure 18 Entity Relationship Diagram

## Data Dictionary

Figure 19 Data Dictionary Part 1

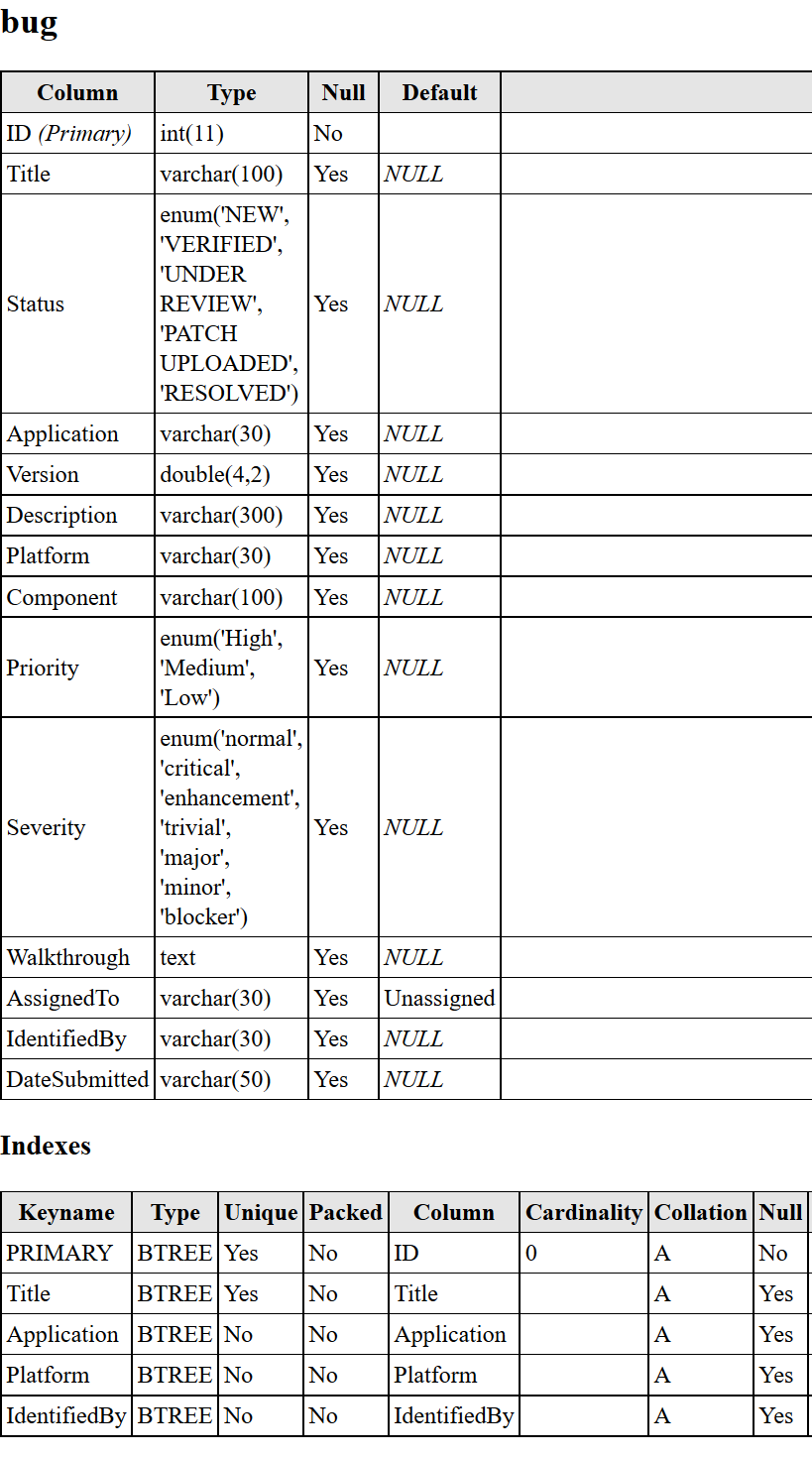


Figure 20 Data Dictionary Part 2

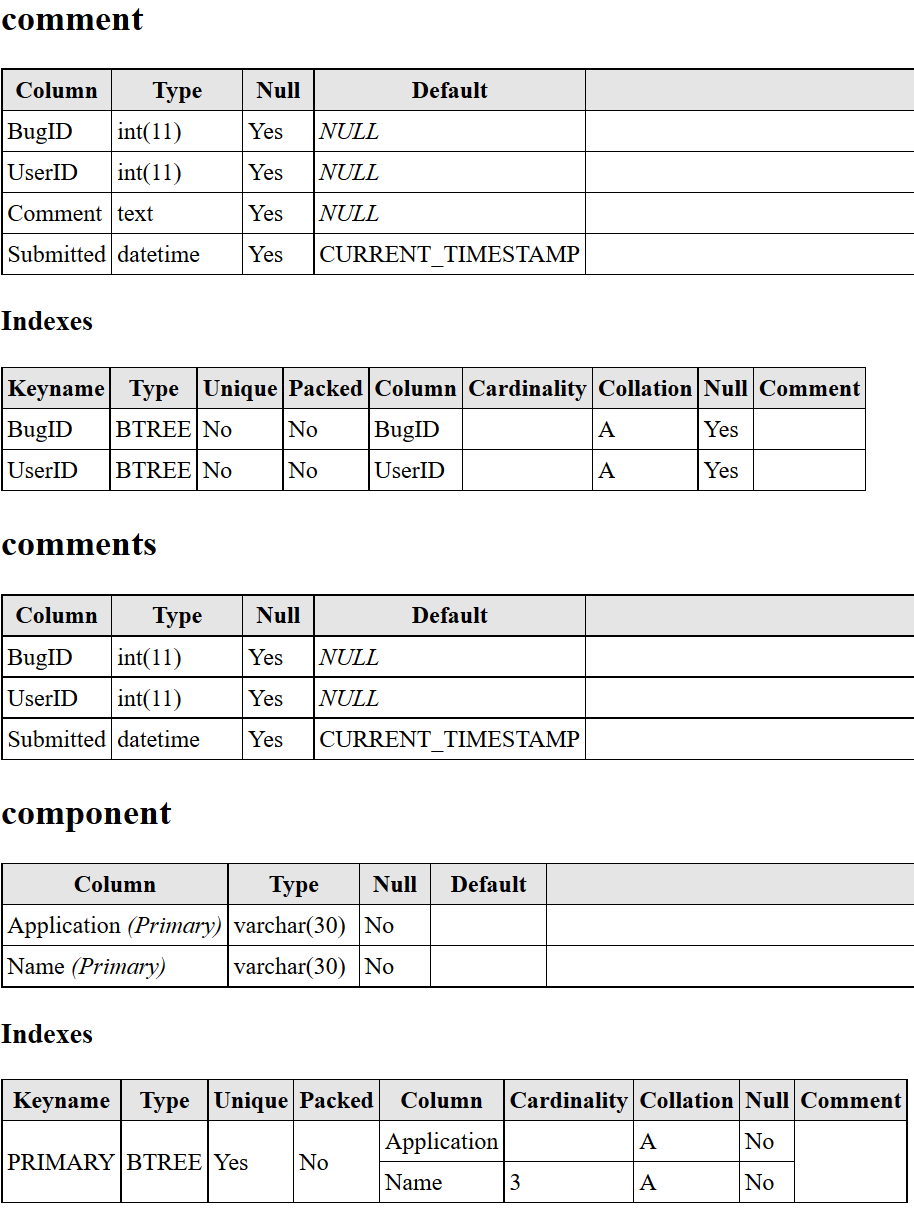


Figure 21 Data Dictionary Part 3

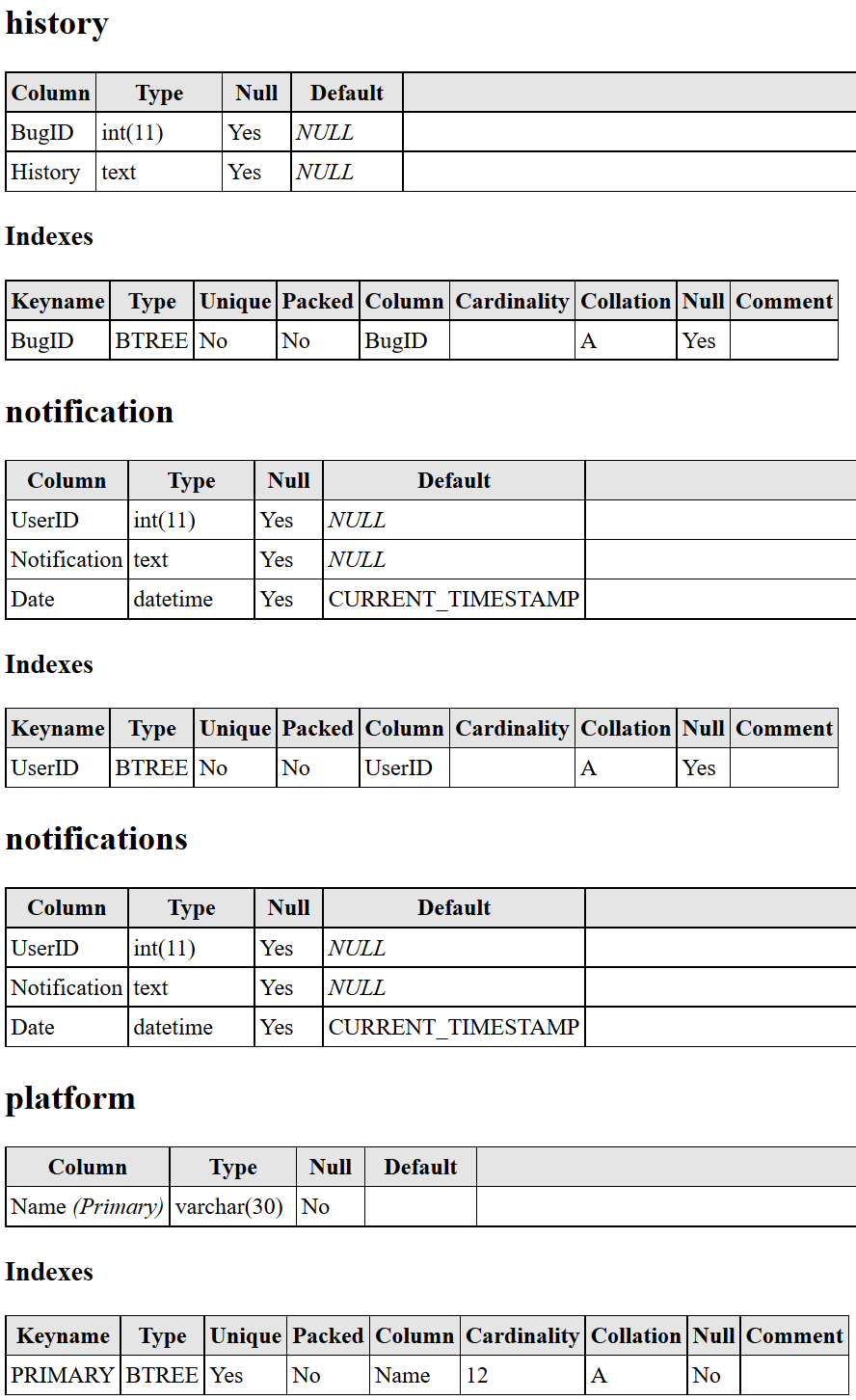


Figure 22 Data Dictionary Part 4

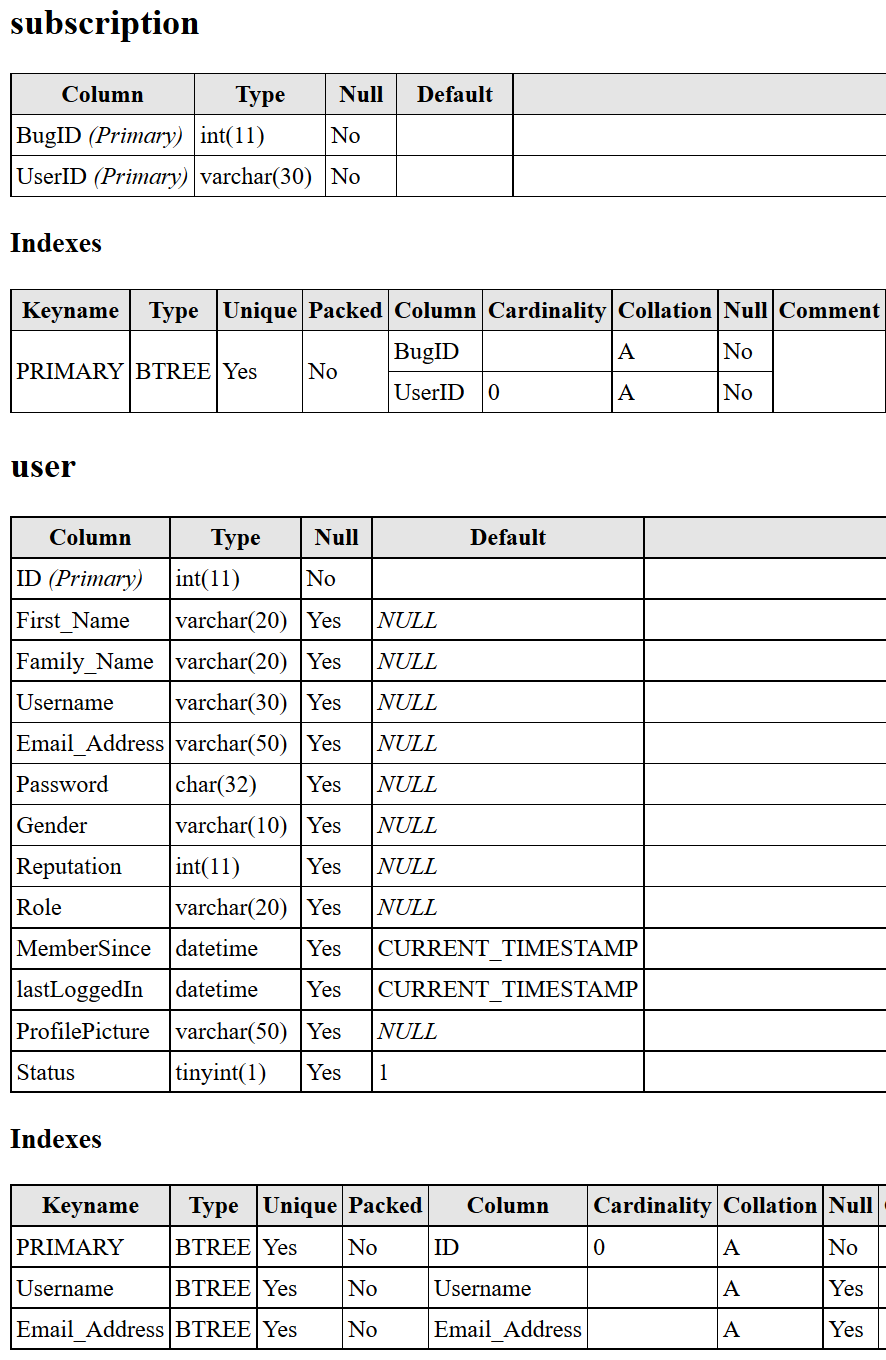


Figure 23 Data Dictionary Part 5

# User Interface Design

## Login Interface

Figure 24 Login Interface Design

## Main Interface / Search Bug Interface

Figure 25 Main Window interface Design

## Add Bug Interface

Figure 26 Add Bug Interface Design

## Profile Interface

Figure 27 Profile Interface Design

## Search User Interface

Figure 28 Search User Interface Design

## Bug Interface

Figure 29 Bug Interface Deign

Figure 30 Report Interface Design

## Report Interface