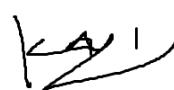
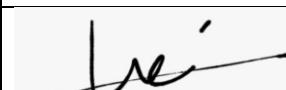
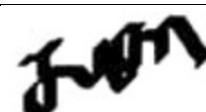

School of Computing and Information Technology

CSIT314

Software Development Methodologies

Group Assignment

(Bug Tracking System)

Team 7268				
Name	UOW ID	Contribution (More Details shown in each Sprints)	Individual Contribution	Signature
Ng Ming Yao (Group Leader)	6342930	- Programming - CICD - DDD - Unit Testing	100%	
Neo Zhi Kai	6342954	- All documentations - All diagrams - Ethical Considerations	100%	
Mallah Rahul Premchand	6344732	- Programming - BCE Diagrams	100%	
Lim Li Shan	6344902	- All documentations - All diagrams	100%	
Liu Zihan	6344781	- Triager Diagrams - User Manual - Demo Video - Bug Report	100%	
Chong Jun Wei	5710911	- Functional Testing	70%	
Ivan Lee You Qing	6347034	- Create Bug Report	35%	

1 Contents

1.	Gantt Chart.....	6
2.	Ethical Consideration and discussions	8
3.	CI/CD	10
4.	Sprint 1.....	11
4.1	User Stories and Diagrams.....	11
4.1.1	Login (User Story #1).....	11
4.1.2	Logout (User Story #7)	14
4.2	Class Diagram.....	16
4.3	Test Cases.....	17
4.3.1	Unit Test Case	17
4.3.2	Functional Test Case	19
4.4	Evidence of Use of Methodologies	20
4.4.1	Meeting Minute	20
4.4.2	Sprint Planning Meeting.....	21
4.4.3	Taiga (End of Sprint).....	22
4.4.4	Sprint Review Meeting.....	22
4.4.5	Sprint Retrospective Meeting	23
4.4.6	Member Contribution	24
5.	Sprint 2	25
5.1	User Stories and Diagrams.....	25
5.1.1	Report Bug (User Story #9)	25
5.1.2	Assign Bug (User Story #11)	28
5.1.3	View Bug Report (User Story #241)	31
5.2	Class Diagram.....	34
5.3	Test Cases.....	35
5.3.1	Functional Test Case	35
5.4	Evidence of Use of Methodologies	37
5.4.1	Meeting Minute	37
5.4.2	Sprint Planning Meeting.....	38
5.4.3	Taiga (End of Sprint).....	39
5.4.4	Sprint Review Meeting.....	40
5.4.5	Sprint Retrospective Meeting	40
5.4.6	Member Contribution	41
6	Sprint 3	42
6.1	User Stories and Diagrams.....	42

6.1.1	View Assigned Bug (User Story #18)	42
6.1.2	Submit Fixed Bug (User Story #19)	45
6.1.3	Input Test Result (User Story #22)	48
6.1.4	Set Bug as Invalid (User Story #13)	51
6.1.5	Set Bug as Solved (User Story #14)	54
6.1.6	Provide Comments (User Stories #8)	57
6.2	Class Diagram.....	60
6.3	Test Cases.....	61
6.3.1	Functional Test Case	61
6.4	Evidence of Use of Methodologies	63
6.4.1	Meeting Minute	63
6.4.2	Sprint Planning Meeting.....	64
6.4.3	Taiga (End of Sprint).....	64
6.4.4	Sprint Review Meeting.....	66
6.4.5	Sprint Retrospective Meeting	66
6.4.6	Member Contribution.....	67
7	Sprint 4	68
7.1	User Stories and Diagrams.....	68
7.1.1	Search Bug Through Keyword (User Story #3).....	68
7.1.2	Search Bug Through Titles (User Story #6)	71
7.1.3	Search Bug Through Assignee (User Stories #4)	74
7.1.4	Search Bug Through Reporter Name (User Story #5)	77
7.1.5	Search Bug Through Bug ID (User Story #2).....	80
7.2	Class Diagram.....	83
7.3	Test Cases.....	84
7.3.1	Functional Test Case	84
7.4	Evidence of Use of Methodologies	85
7.4.1	Meeting Minute	85
7.4.2	Sprint Planning Meeting.....	86
7.4.3	Taiga (End of Sprint).....	86
7.4.4	Sprint Review Meeting.....	88
7.4.5	Sprint Retrospective Meeting	88
7.4.6	Member Contribution.....	89
8	Sprint 5	90
8.1	User Stories and Diagrams.....	90
8.1.1	Generate Reported Bug Report (User Story #15)	90

8.1.2	Generate Resolved Bug Report (User Story #16).....	93
8.2	Class Diagram.....	96
8.3	Test Cases.....	97
8.3.1	Functional Test Case	97
8.4	Evidence of Use of Methodologies	98
8.4.1	Meeting Minute	98
8.4.2	Sprint Planning Meeting.....	99
8.4.3	Taiga (End of Sprint).....	99
8.4.4	Sprint Review Meeting.....	100
8.4.5	Sprint Retrospective Meeting	100
8.4.6	Member Contribution	101
9	Data Persistence Diagram.....	102
10	Class Diagram.....	103
11	Data-Driven Development	104
12	Project Closure.....	105

1. Gantt Chart

PROJECT TITLE:	Bug Tracking System																															
Project Start:	Thu, 01-Oct-2020																															
Display month:	0																															
Employee name:	Rahul	Ming Yao	Zhi Kai																													
No. of task:	21	22	41																													
Workload (%):	15.91	16.67	3106																													
Employee name:	Li Shan	Zi Han	Jun Wei	Ivan																												
No. of task:	24	10	6	8																												
Workload (%):	18.18	7.58	4.55	6.0606061																												

2. Ethical Consideration and discussions

Introduction

The purpose of this is to overview ethical issues that should be considered when implementing this software methodology. Ethical considerations help to determine the difference between acceptable and unacceptable behaviours. The ethical standards will prevent against fabrication or falsifying of data and thus promoting the pursuit of knowledge and truth which is the primary goal. Ethical behaviour is also critical for collaborative work because it encourages an environment of trust, accountability, and mutual respect among the team.

Ethics Issues

Privacy

The handling, storing, and sharing of user data can only be done only under the circumstances and for the purposes that the user sets.

Accountability

Who should be held responsible for the harm caused by the software.

Transparency

The transparent decision-making procedures of intelligent systems and publicly available ethics policies by software development organizations.

Work ethics

The decisions on which bugs to fix and how quickly it can be fixed and ensuring quality of the code before official release of the system.

Ethical Considerations

According to Association for Computing Machinery (ACM) Code of Ethics for Software Development, we will follow the code where every software professional has obligations to, in the list shown below:

1. PUBLIC/SOCIETY

Team must adhere to ethical standards in order for the public to support and believe in them. The public wants to be assured that the team followed the appropriate guidelines for issues such as safety, human rights, compliance with the law, conflicts of interest and so on.

2. CLIENT AND EMPLOYER

Team must provide service in their areas of competence and be honest and forthright about any limitations of their experience and education. The team must not use software that is obtained illegally and not to keep any confidential information for their own personal interest.

3. PRODUCT

Team must strive for high quality, acceptable cost, and a reasonable schedule and to ensure there are adequate testing, debugging and review of software and related documents the team is working on. All this to ensure the products and related modifications meet the highest professional standards possible.

4. JUDGEMENT

Team will only endorse documents under their supervision and disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided. The team must maintain integrity and independence in their professional judgment.

5. MANAGEMENT

Team must ensure good management for this project they are working on and ensure there is a fair agreement concerning ownership of any work contributed by each individual member.

6. PROFESSION

Team must take responsibility for detecting, correcting, and reporting errors in software and associated documents on which they work.

7. COLLEAGUES

Members in the team must be fair and supportive of their colleagues. Each member should credit fully the work of others and refrain from taking undue credit. They should also play a part to review the work of others in an objective, candid, and properly-documented way.

8. SELF

Every member in the team should participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Additional notes

If the team agrees to make modifications to the program, it should do so with the understanding that the time required will be sufficient to perform a thorough check out. This is to prevent premature outcome as poorly designed effort could clearly create unforeseen problems that might cost more time in the future timeframe.

Another ethical obligation of the team is to understand the requirements and the risks, to apply their best technical judgement and to candidly discuss the options with everyone in the team. Agreeing to perform a task poorly simply because another team member requested is not acceptable.

3. CI/CD

We have tried to develop the Continuous Integration/Continuous Deployment (CI/CD) on the GitLab platform, but we faced difficulty and we are unable to solve the error. First, we have tried to follow the steps on the tutorial, but an error occurred. Therefore, we followed what our lecturer teaches us in the lecture, but we still face error of multiple packages do not exist. To try to resolve it, we tried to include the package's path in the .yml file just like how the lecturer teach us to include the junit's path. We have tried different solution that we have found online and few of our team members also tried doing it on different computers but the error still persists.

The screenshot shows the GitLab web interface for a project named 'TEAM7268'. The sidebar on the left lists various project sections: Project overview, Repository (selected), Files, Commits, Branches, Tags, Contributors, Graph, Compare, Issues (0), Merge Requests (0), CI / CD (selected), Security & Compliance, Operations, and Collapse sidebar. The main content area displays a table of files with their last commit details:

Name	Last commit	Last update
Comments	Add file	1 hour ago
build	Add	57 minutes ago
dist	Add	57 minutes ago
jcalendar-1.4	Add files	2 minutes ago
lib	Add files	9 minutes ago
noproject	Add files	2 minutes ago
src/BugTracker	Add file	1 hour ago
test/BugTracker	Add file	1 hour ago
.gitlab-ci.yml	Update .gitlab-ci.yml	1 minute ago
BugReport.txt	Add file	1 hour ago
accounts.txt	Add file	1 hour ago
bug10000.txt	Add file	1 hour ago
build.xml	Add file	1 hour ago
manifest.mf	Add file	1 hour ago

Below this, there is a table showing the CI pipeline stages:

Status	Pipeline	Triggerer	Commit	Stages
failed	#217097936 latest	●	⚡ master -> 73db3e1b Update .gitlab-ci.yml	⌚ 00:00:31 ⌚ 35 seconds ago
failed	#217097549	●	⚡ master -> 99bf9886 Merge origin/master	⌚ 00:00:28 ⌚ 57 seconds ago
failed	#217095567	●	⚡ master -> f5845be1 Update .gitlab-ci.yml	⌚ 00:00:34 ⌚ 3 minutes ago
failed	#217092091	●	⚡ master -> 83190b37 Update .gitlab-ci.yml	⌚ 00:00:33 ⌚ 6 minutes ago
failed	#217090964	●	⚡ master -> 802b7db2 Merge origin/master	⌚ 00:00:39 ⌚ 7 minutes ago
failed	#217087593	●	⚡ master -> df638bcc Update .gitlab-ci.yml	⌚ 00:00:31 ⌚ 11 minutes ago

Status	Pipeline	Triggerer	Commit	Stages
failed	#217097936 latest	●	⚡ master -> 73db3e1b Update .gitlab-ci.yml	⌚ 00:00:31 ⌚ 35 seconds ago
failed	#217097549	●	⚡ master -> 99bf9886 Merge origin/master	⌚ 00:00:28 ⌚ 57 seconds ago
failed	#217095567	●	⚡ master -> f5845be1 Update .gitlab-ci.yml	⌚ 00:00:34 ⌚ 3 minutes ago
failed	#217092091	●	⚡ master -> 83190b37 Update .gitlab-ci.yml	⌚ 00:00:33 ⌚ 6 minutes ago
failed	#217090964	●	⚡ master -> 802b7db2 Merge origin/master	⌚ 00:00:39 ⌚ 7 minutes ago
failed	#217087593	●	⚡ master -> df638bcc Update .gitlab-ci.yml	⌚ 00:00:31 ⌚ 11 minutes ago

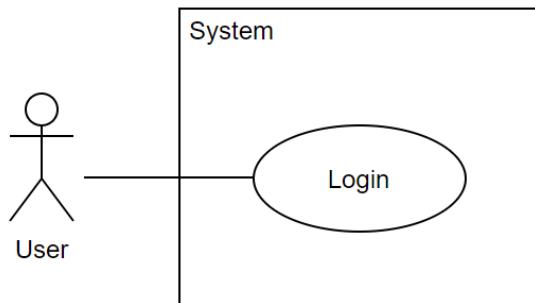
4. Sprint 1

4.1 User Stories and Diagrams

4.1.1 Login (User Story #1)

As a user, I want to be able to log in to my account using my id and password to the system so that I can do my intended work in it.

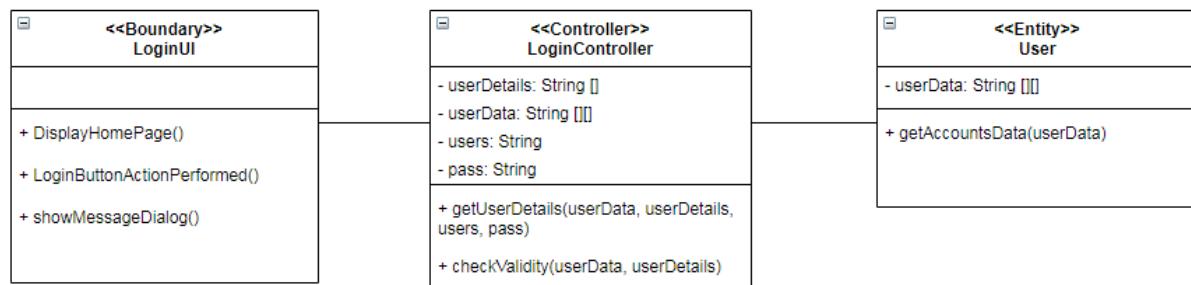
Use Case Diagram



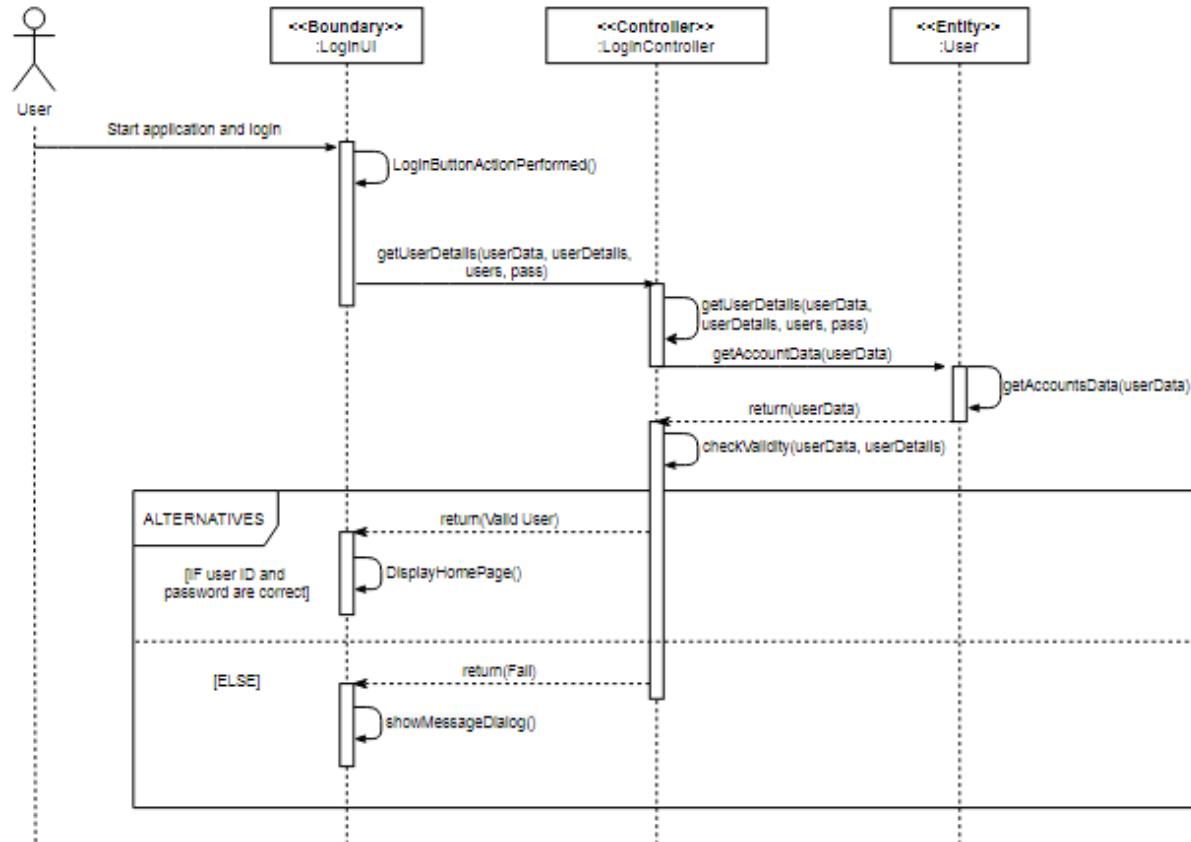
Use Case Description

Use Case Name: Login	ID: 1
Stakeholders and goals: User want to be able to login	
Description: A user wants login to the system using his user ID and password	
Actors: User	
Trigger: A user launch the system and login using his user ID and password	
Normal flow: <ol style="list-style-type: none">1. System displays a login page2. User enters user ID and password3. The system verifies the ID and password4. User gain access to the system5. System displays menu according to the user type (normal user/ triager/ developer/ reviewer)	
Sub flows: None	
Alternatives/Exceptional flows: 4a. Login Unsuccessful: System displays wrong username or password	

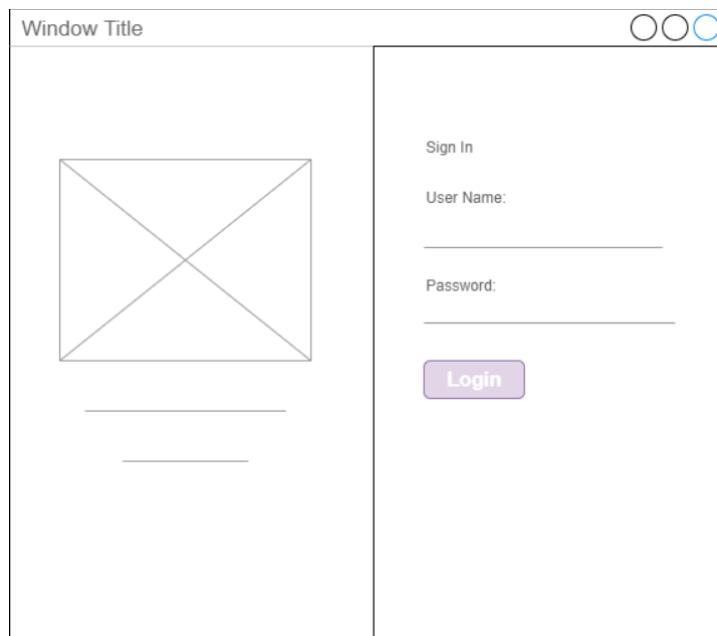
BCE Class Diagram



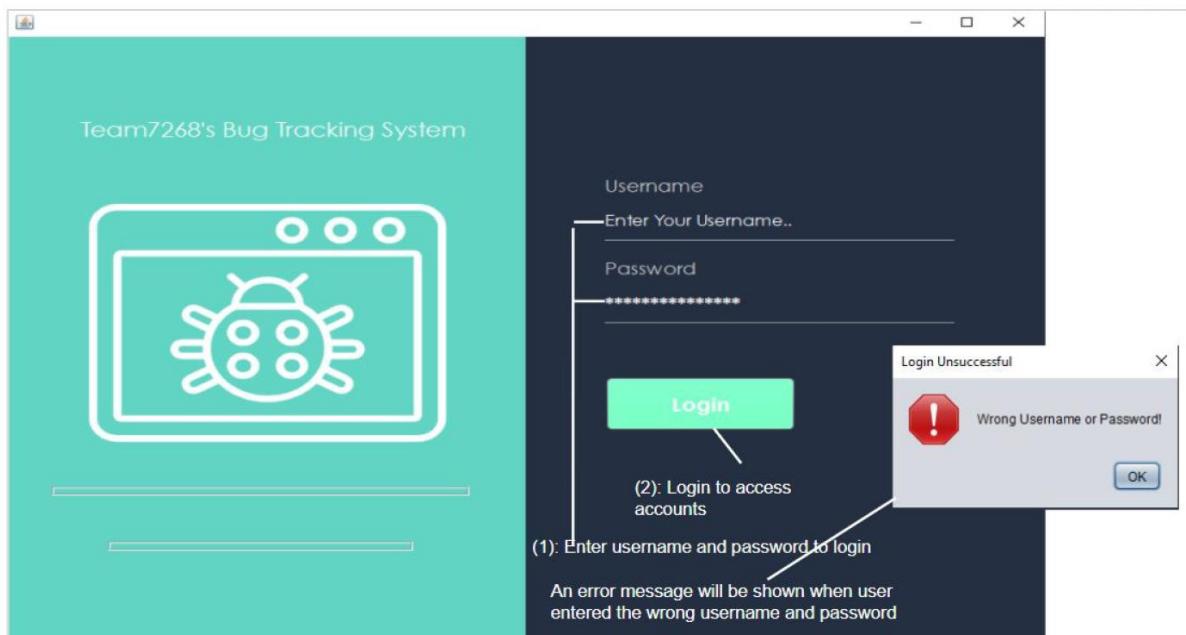
Sequence Diagram



Wireframe



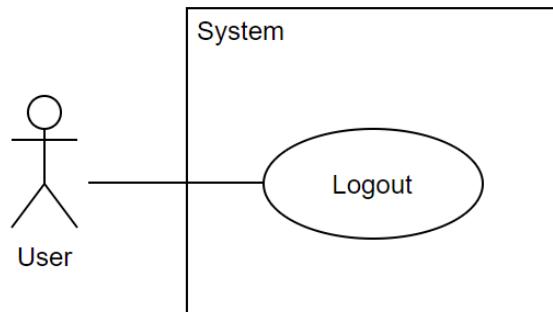
GUI



4.1.2 Logout (User Story #7)

As a user, I want to be able to log out of my account so that I can close the system in a proper manner.

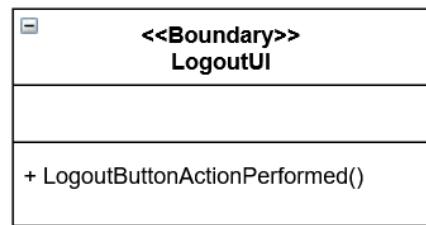
Use Case Diagram



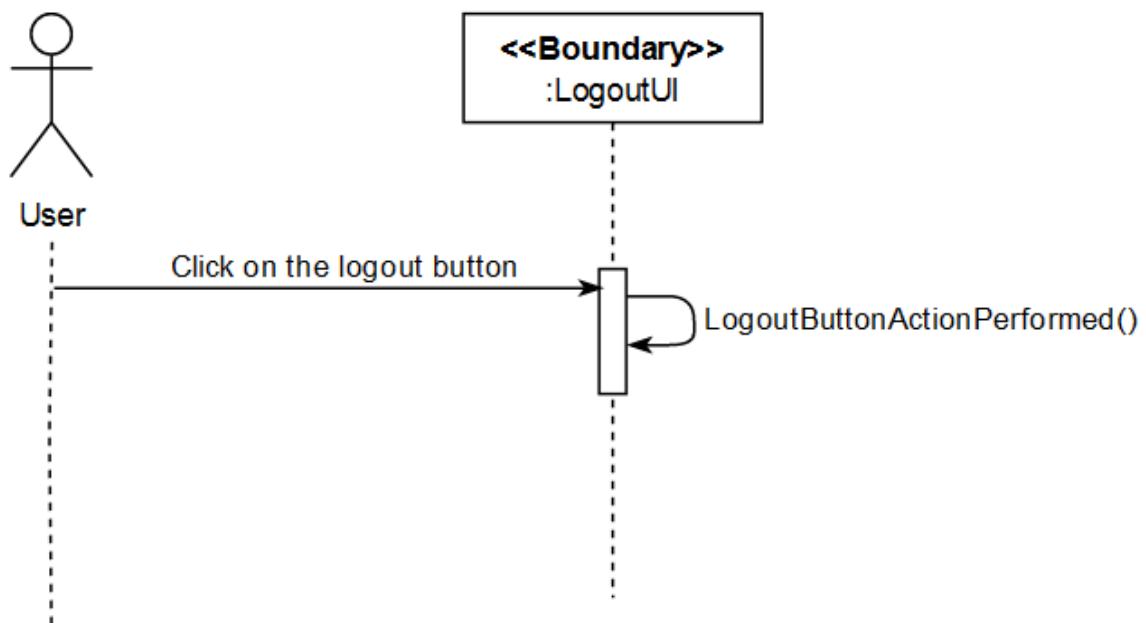
Use Case Description

Use Case Name: Logout	ID: 7
Stakeholders and goals: User want to be able to logout of the system	
Description: A user wants to logout of the system	
Actors: User	
Trigger: A user click on the logout button to logout of the system	
Preconditions: User must be logged in to the system	
Normal flow: <ol style="list-style-type: none">1. The user clicks on the logout button2. System ends all current user's session3. System redirects user to the Login Page4. End	
Sub flows: None	
Alternatives/Exceptional flows: None	

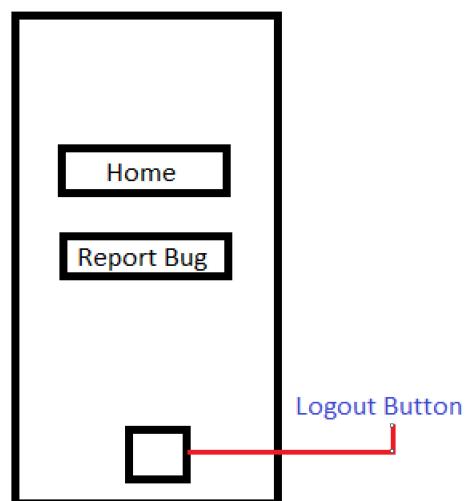
BCE Class Diagram



BCE Sequence Diagram



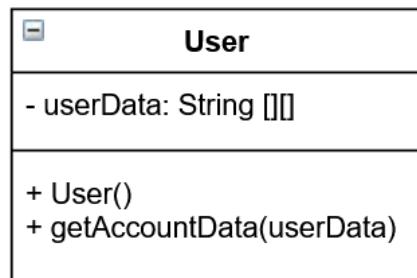
Wireframe



GUI



4.2 Class Diagram



4.3 Test Cases

4.3.1 Unit Test Case

Test Code

The image shows two side-by-side Java code editors. Both editors have a title bar 'Start Page' and a tab labeled 'LoginFormTest.java'. The top editor (lines 1-31) contains the class definition and setup/teardown methods for the unit test:

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package BugTracker;
7
8 import org.junit.AfterClass;
9 import org.junit.BeforeClass;
10 import org.junit.Test;
11 import static org.junit.Assert.*;
12
13 /**
14  *
15  * @author malla
16  */
17 public class LoginFormTest {
18
19     public LoginFormTest() {
20     }
21
22     @BeforeClass
23     public static void setUpClass() {
24     }
25
26     @AfterClass
27     public static void tearDownClass() {
28     }
29
30     /**
31      * Test of checkValidity method, of class LoginForm.
```

The bottom editor (lines 34-88) contains the implementation of the test methods:

```
34     public void testCheckValidity() {
35         System.out.println("Checking User Validity");
36         System.out.println("-----");
37         String[][] accountDataDeveloper = {"D12345","qweasd","David White"};
38         String[][] accountDataUser = {"U12345","qweasd","Walter Green"};
39         String[][] accountDataTriager = {"T12345","qweasd","Barry White"};
40         String[][] accountDataReviewer = {"R12345","qweasd","John Davis"};
41         String[] userDetailsServiceDeveloper = {"D12345","qweasd"};
42         String[] userDetailsServiceUser = {"U12345","qweasd"};
43         String[] userDetailsServiceTriager = {"T12345","qweasd"};
44         String[] userDetailsServiceReviewer = {"R12345","qweasd"};
45
46         System.out.println("Checking Developer...");
47         LoginForm instanceDev = new LoginForm();
48         String expResultDev = "Valid UserDavid White";
49         String resultDev = instanceDev.checkValidity(accountDataDeveloper, userDetailsServiceDeveloper);
50         assertEquals(expResultDev, resultDev);
51         System.out.println("Developer checked, Login Successful...\n");
52
53         System.out.println("Checking User...");
54         LoginForm instanceUser = new LoginForm();
55         String expResultUser = "Valid UserWalter Green";
56         String resultUser = instanceUser.checkValidity(accountDataUser, userDetailsServiceUser);
57         assertEquals(expResultUser, resultUser);
58         System.out.println("User checked, Login Successful...\n");
59
60         System.out.println("Checking Triager...");
61         LoginForm instanceTri = new LoginForm();
62         String expResultTri = "Valid UserBarry White";
63         String resultTri = instanceTri.checkValidity(accountDataTriager, userDetailsServiceTriager);
64         assertEquals(expResultTri, resultTri);
65
66         System.out.println("Triager checked, Login Successful...\n");
67
68         System.out.println("Checking Reviewer...");
69         LoginForm instanceRev = new LoginForm();
70         String expResultRev = "Valid UserJohn Davis";
71         String resultRev = instanceRev.checkValidity(accountDataReviewer, userDetailsServiceReviewer);
72         assertEquals(expResultRev, resultRev);
73         System.out.println("Reviewer checked, Login Successful...\n");
74         // TODO review the generated test code and remove the default call to fail.
75
76     /**
77      * Test of main method, of class LoginForm.
78     */
79     @Test
80     public void testMain() {
81         String[] args = null;
82         LoginForm.main(args);
83         // TODO review the generated test code and remove the default call to fail.
84         //fail("The test case is a prototype.");
85     }
86
87 }
88
```

Test Fail (Before Function Implementation)

The screenshot shows a 'Test Results' window for 'BugTracker.LoginFormTest'. The status bar at the top says 'Tests passed: 50.00 %'. Below it, a message states '1 test passed, 1 test caused an error. (0.412 s)'. A tree view shows a single failed test: 'BugTracker.LoginFormTest Failed' under 'testCheckValidity caused an ERROR'. The right pane displays log output:

```
Checking User Validity
-----
Checking Developer...

```

Test Pass (After Function Implementation)

The screenshot shows a 'Test Results' window for 'BugTracker.LoginFormTest'. The status bar at the top says 'Tests passed: 100.00 %'. Below it, a message states 'Both tests passed. (0.442 s)'. The tree view shows both tests have passed. The right pane displays log output:

```
Checking User Validity
-----
Checking Developer...
Developer checked, Login Successful...

Checking User...
User checked, Login Successful...

Checking Triager...
Triager checked, Login Successful...

Checking Reviewer...
Reviewer checked, Login Successful...
```

4.3.2 Functional Test Case

Login

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Check login function						
Created By:	Li Shan						
Tested By:	Rahul						
Tested On:	15/10/2020						
Preconditions:	None						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Launch the program		Bug Tracking System would launch and login page would appear	Login page appears on screen	Pass	Login Page appears center of the screen.	
USER-01-2-1	Unsuccessful login - Empty cell in User ID / Password	(1) Click on Login button	A pop up message would appear warning user that the fields cannot be empty	LoginUnsuccessful message dialog appears on screen saying "Fields cannot be empty"	Pass	Dialog box requires you to click ok	
USER-01-2-2	Unsuccessful login - Empty cell in Password	(1) Input a valid User ID *refer to test data* (2) Click on Login button	A pop up message would appear warning user that the fields cannot be empty	Login Unsuccessful message dialog appears on screen saying "Fields cannot be empty"	Pass	Dialog box requires you to click ok	
USER-01-2-3	Unsuccessful login - Empty cell in User ID	(1) Input Password = abcdefg (2) Click on Login button	A pop up message would appear warning user that the fields cannot be empty	Login Unsuccessful message dialog appears on screen saying "Fields cannot be empty"	Pass	Password field text are hidden.	
USER-01-2-4	Unsuccessful login - Invalid User ID	(1) Input User ID = U123456 (2) Input Password = abcdefg (3) Click on Login	A pop up message would appear warning user that the User ID or Password is invalid	Login Unsuccessful message dialog appears on screen saying "Invalid username or password".	Pass	Dialog box requires you to click ok	
USER-01-2-5	Unsuccessful login - Incorrect Password	(1) Input a valid User ID *refer to test data* (2) Input Password = abcdefg (3) Click on Login button	A pop up message would appear warning user that the User ID or Password is invalid	Login Unsuccessful message dialog appears on screen saying "Invalid username or password".	Pass	Dialog box requires you to click ok	
USER-01-2-6	Successful login - Normal User	(1) Input a valid Normal User ID *refer to test data* (2) Input valid Password *refer to test data* (3) Click on Login button	Login successful. The system would display the menu for Normal User	Normal User home menu page is displayed on the screen.	Pass	The page opens instantaneously upon login	
USER-01-2-7	Successful login - Traiger	(1) Input a valid Traiger User ID *refer to test data* (2) Input valid Password *refer to test data* (3) Click on Login button	Login successful. The system would display the menu for Traiger	Traiger home menu page is displayed on the screen.	Pass	Noticeably longer to open compared to normal user.	
USER-01-2-8	Successful login - Developer	(1) Input a valid Developer User ID *refer to test data* (2) Input valid Password *refer to test data* (3) Click on Login button	Login successful. The system would display the menu for Developer	Developer home menu page is displayed on the screen.	Pass	Opens instantly upon login	
USER-01-2-9	Successful login - Reviewer	(1) Input a valid Reviewer User ID *refer to test data* (2) Input valid Password *refer to test data* (3) Click on Login button	Login successful. The system would display the menu for Reviewer	Reviewer home menu page is displayed on the screen.	Pass	Opens instantly upon login	

Logout

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Check logout function						
Created By:	Li Shan						
Tested By:	Ming Yao						
Tested On:	15/10/2020						
Preconditions:	User must be logged in to the system						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-3-1	Successful logout - Normal User	(1) Click on Logout button	The system would end the user current session and redirect to the Login page	Successful logout from the system and the login page showed up.	Pass		
USER-01-3-2	Successful logout - Traiger	(1) Click on Logout button	The system would end the user current session and redirect to the Login page	Successful logout from the system and the login page showed up.	Pass		
USER-01-3-3	Successful logout - Developer	(1) Click on Logout button	The system would end the user current session and redirect to the Login page	Successful logout from the system and the login page showed up.	Pass		
USER-01-3-4	Successful logout - Reviewer	(1) Click on Logout button	The system would end the user current session and redirect to the Login page	Successful logout from the system and the login page showed up.	Pass		

4.4 Evidence of Use of Methodologies

4.4.1 Meeting Minute

Project Name: Bug tracking system

Meeting Objective: Discussions with product owner and Kick-off Sprint planning

Date: 1st October 2020

Attendee
Ng Ming Yao
Lim Li Shan
Neo Zhi Kai
Liu Zihan
Mallah Rahul Premchand
Chong Jun Wei

Agenda:

- Introduction of attendees
- Review of project requirements
- Discussion on user stories
- Define what can be delivered in Sprint 1
- Decide on the goal for Sprint 1
- Create Product Backlog and Sprint 1 Backlog

S/No	Item	Action Item	Due Date
1	Distribution of tasks	Task distributed by MingYao	1 st October 2020
2	Complete user stories	To be done by team	3 rd October 2020
3	Decide how to achieve Sprint 1 goal	Discussed by team	3 rd October 2020
4	Create product backlog on taiga	Done by team	4 th October 2020
5	Create sprint backlog	Done by Zhi Kai	4 th October 2020
6	Complete all tasks on Sprint 1	Done by team	7 th October 2020

Date and time of next meeting:

Sprint 2: 8th October 2020

Meeting minutes 1 prepared by Zhi Kai

4.4.2 Sprint Planning Meeting

Sprint prioritization

- Analyse and evaluate product backlog, product backlog items are prioritized
- Estimation done for each user story.
- Sprint goal
 - To complete software system with the login and logout functionalities for all user

Taiga - Product Backlog (New Items Added)

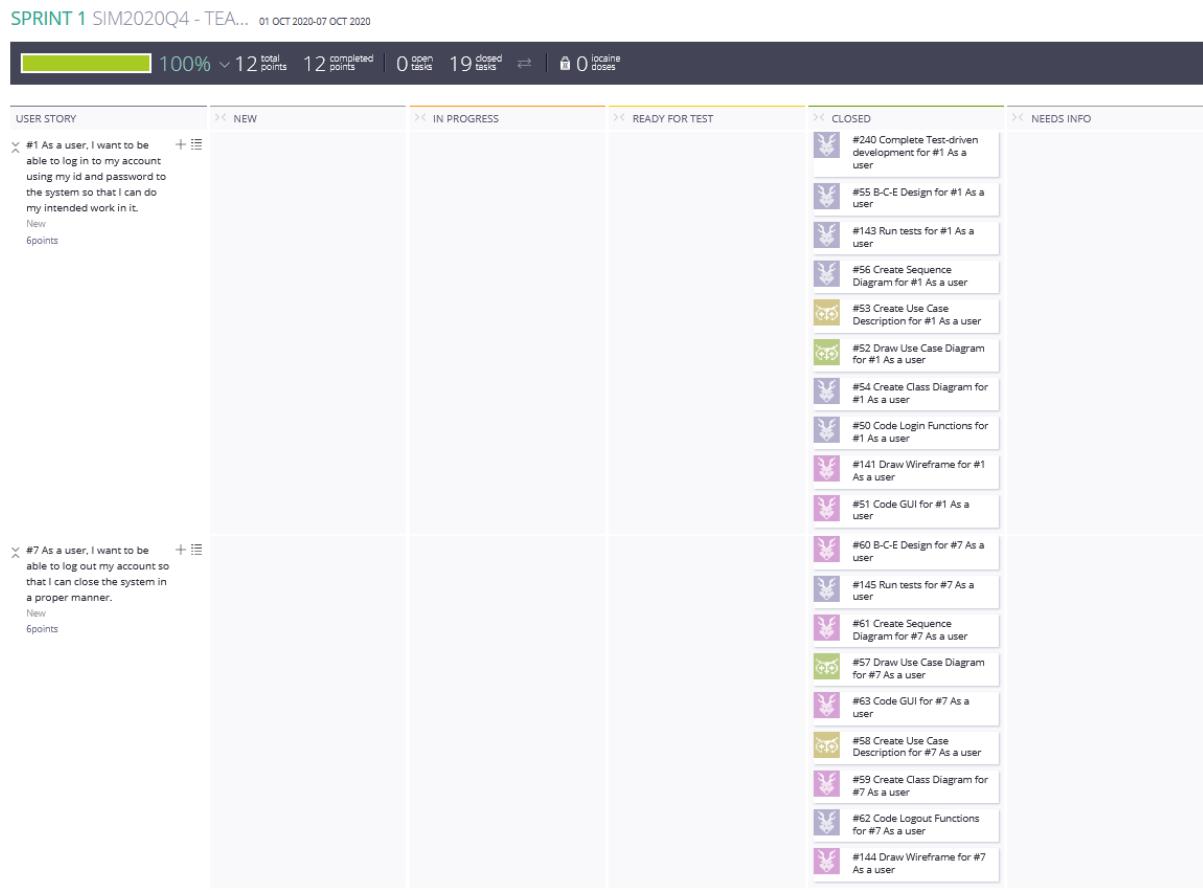
BACKLOG SIM2020Q4 - TEAM7268

0% 156 defined 0 closed 0 points / sprint

SHOW FILTERS SHOW TAGS + ADD A NEW USER STORY

Votes	User Stories	Status	Points
0	#1 As a user, I want to be able to log in to my account using my id and password to the system so that I can do my intended work in it.	New ↗	6
0	#7 As a user, I want to be able to log out my account so that I can close the system in a proper manner.	New ↗	6
0	#9 As a user, I want to be able to report a bug with its issue description and bug summary so that the triager can understand the problem and properly assign the bug to the developer to resolve the bug.	New ↗	12
0	#10 As a triager, I want to be able to view new bugs reported in the default page so that I can assign the bugs efficiently to the developers.	New ↗	12
0	#11 As a triager, I want to be able to assign bugs to different developers so that they can work to resolve it.	New ↗	9
0	#12 As a triager, I want to be able to fix the severity of bugs so that the developer can know the degree of impact that a defect has on the system.	New ↗	5
0	#17 As a triager, I want to be able to fix the priority of bugs so that I can prioritize the order of severity which has impacted the system in order for the developer to handle higher prioritized bugs.	New ↗	5
0	#18 As a developer, I want to be able to check for my assigned task's status and problem reported so that I can begin fixing it.	New ↗	12
0	#19 As a developer, I want to be able to submit my assigned task so that it can be tested by the reviewers.	New ↗	9
0	#22 As a reviewer, I want to be able to input my testing result so that triager could mark the bug as solved.	New ↗	7
0	#13 As a triager, I want to be able to set a bug as invalid so that invalid bugs can be ignored.	New ↗	6
0	#14 As a triager, I want to be able to set the status of a bug as solved so that it can show that the bug has been resolved.	New ↗	7
0	#3 As a user, I want to be able to search a bug through keywords so that I can track down a specific bug's ID, description, summary, reported date, assigned developer and current status.	New ↗	6
0	#4 As a user, I want to be able to search a bug through titles so that I can track down a specific bug's ID, description, summary, reported date, assigned developer and current status.	New ↗	6
0	#6 As a user, I want to be able to search using an assignee name so that I can see all the bugs assigned or resolved by this assignee.	New ↗	6
0	#5 As a user, I want to be able to search using a reporter name so that I can see all the bugs reported by this reporter.	New ↗	6
0	#2 As a user, I want to be able to search a bug using the bug ID so that I can track down a specific bug's description, summary, reported date, assigned developer and current status.	New ↗	6
0	#15 As a triager, I want to be able to generate a report that shows total numbers of bugs reported in a month and reported by whom so that I can see the performance of reporters.	New ↗	7
0	#16 As a triager, I want to be able to generate a report that shows the total number of bugs resolved in a week and resolved by whom so that I can see the performance of developers.	New ↗	7
0	#8 As a user, I want to be able to provide comments so that I can participate in the discussion of a certain bug.	New ↗	8
0	#20 As a developer, I want to be able to fill in my expertise so that I am able to be assigned to fixed bugs that are under my expertise. (low priority)	New ↗	4
0	#21 As a developer, I want to be able to view the total number of bugs fixed by me so that I can know how I performed. (low priority)	New ↗	4

4.4.3 Taiga (End of Sprint)



4.4.4 Sprint Review Meeting

Work accomplished during Sprint 1:

- Review product requirements from Terence
- Team created user stories for product backlog in taiga
- User stories added to product backlog
- Team decided on the goal for sprint 1 and select items from product backlog they commit to complete to achieve the sprint's goal
- Sprint 1 backlog is created in taiga, tasks are identified and each is estimated
- Done system functionalities for Login and Logout for all user and tested

4.4.5 Sprint Retrospective Meeting

Work to complete next reporting period:

- Preparation for Sprint 2
- Reprioritize features according to requirements

What went well in the Sprint:

- Team members are able to agree and accept on a similar goal thus making it easier for the project to be carried out
- Team members are good in their respective tasks as they are assigned according to their strength and weakness and therefore the assigned tasks in the sprint is able to proceed smoothly
- Goals and instructions for each individual are clear, team members are aware of the direction we are moving.

What can be improved:

- Spend more time making priorities for the user stories
- Improve in focusing the goals for the sprint and commit to complete to achieve the sprint's goal.

(Discussions) Suggestions/Issues:

- Have to carefully review the priorities of the user stories and the goals we set so to ensure we can complete the required tasks in a sprint on schedule.

4.4.6 Member Contribution

Name:	Task Completed
Ng Ming Yao	<ul style="list-style-type: none"> • Distribution of roles • Prepared user stories and refining • Created user backlog • Analyzed and evaluated product backlog • Completed assigned diagrams • Completed use case description • Estimated user stories points • Completed coding user Login and Logout functionalities • Complete testing for sprint 1 user stories • Complete Sprint 1 unit testings
Lim Li Shan	<ul style="list-style-type: none"> • Prepared user stories and refining • Analyzed and evaluated product backlog • Completed assigned diagrams • Completed use case description • Estimated user stories points • Created test script for sprint 1 user stories
Neo Zhi Kai	<ul style="list-style-type: none"> • Prepared user stories and refining • Created meeting minutes 1 • Created week 1 report • Completed assigned diagrams • Completed use case description • Analyzed and evaluated product backlog • Prepared sprint 1 backlog • Estimated user stories points
Liu Zihan	<ul style="list-style-type: none"> • Prepared user stories draft • Completed assigned diagrams • Completed use case description
Mallah Rahul Premchand	<ul style="list-style-type: none"> • Prepared user stories and refining • Completed assigned diagrams • Completed use case description • Estimated user stories points • Created Wireframe • Completed coding GUI for login and logout • Complete testing for sprint 1 user stories
Chong Jun Wei	<ul style="list-style-type: none"> • Prepare user stories draft • Completed assigned diagrams • Completed use case description
Ivan Lee You Qing	<ul style="list-style-type: none"> • Completed assigned use case diagrams

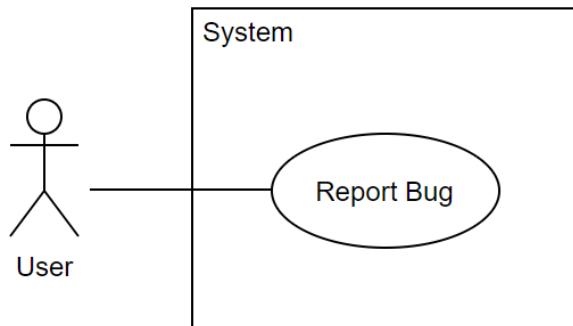
5. Sprint 2

5.1 User Stories and Diagrams

5.1.1 Report Bug (User Story #9)

As a user, I want to be able to report a bug with its title, keyword, issue description and bug summary so that the triager can understand the problem and properly assign the bug to the developer to resolve the bug.

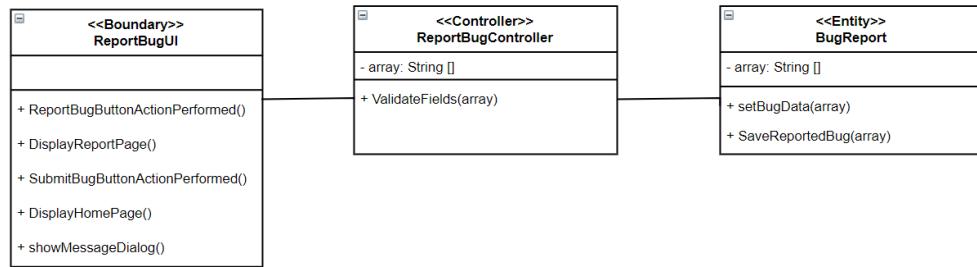
Use Case Diagram



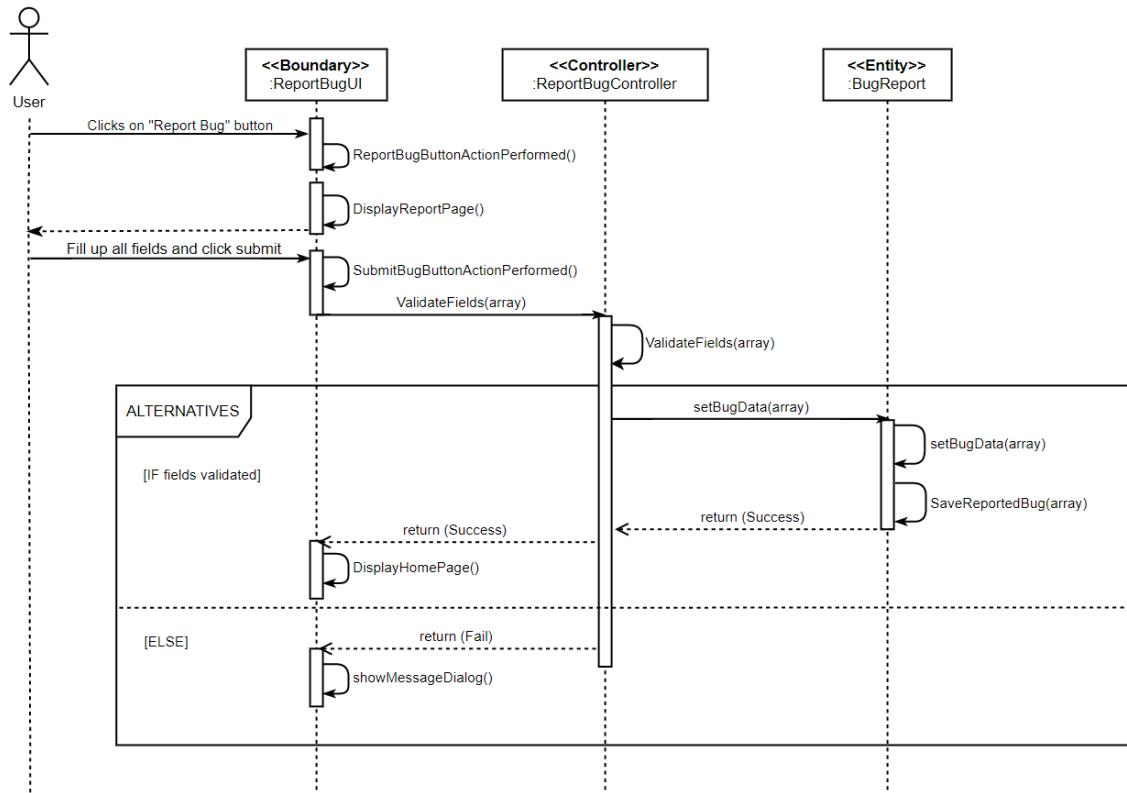
Use Case Description

Use Case Name: Report Bug	ID: 9
Stakeholders and goals: User want to be able to report a bug	
Description: A user report a bug after filling up the bug titles, keyword, description and bug summary	
Actors: User	
Trigger: A user selects the “Report Bug” button, fills up the bug title, keyword, issue description and bug summary and select the “Submit Bug” button	
Preconditions: User must be logged in to the system	
Normal flow: <ol style="list-style-type: none">1. The user clicks the “Report Bug” button2. System directs user to a page to fill up the bug titles, keyword, issue description and summary of the bug3. The user clicks the “Submit Bug” button4. System will automatically assign a Bug ID and current date to the bug report5. System will display the Bug’s ID, Title, Keyword, Date, Description, Summary, Reporter’s name, Assigned Developer (newly assigned bug should show null), Current Status (default: Not Resolved), Priority (default: null), Severity (default: null), Patch Summary, Reviewer and Test Results in the Dashboard (Home Page)6. End	
Sub flows: None	
Alternatives/Exceptional flows: 4a. Unable to submit: System will display fields cannot be empty	

BCE Class Diagram



Sequence Diagram

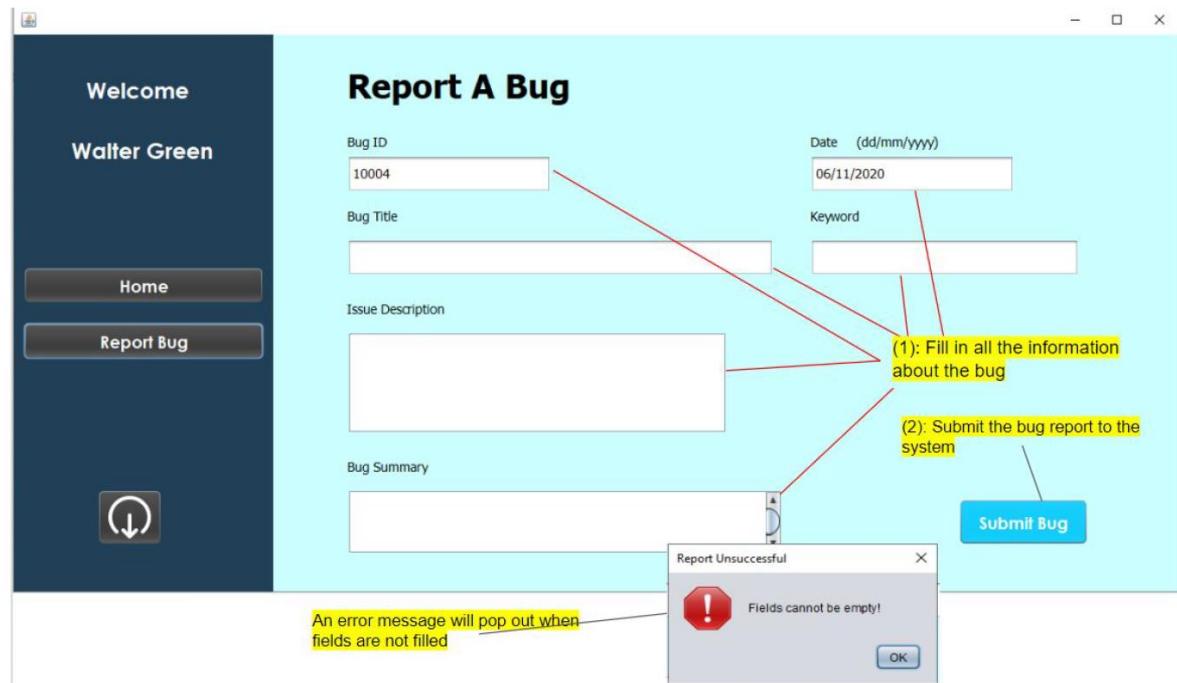


Wireframe

The wireframe shows the layout of the "Report A Bug" form:

- User's name:** Input field.
- Report Bug:** Button.
- Report A Bug:** Main title.
- Bug ID:** Input field.
- Date:** Input field.
- Bug Title:** Input field.
- Keyword:** Input field.
- Issue Description:** Large input area.
- Bug Summary:** Input field.
- Submit Bug:** Button.

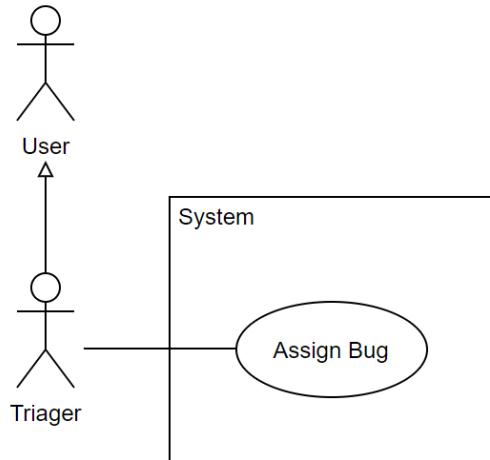
GUI



5.1.2 Assign Bug (User Story #11)

As a triager, I want to be able to set the priority and severity of a bug and assign the bug to different developers so that they can work to resolve it.

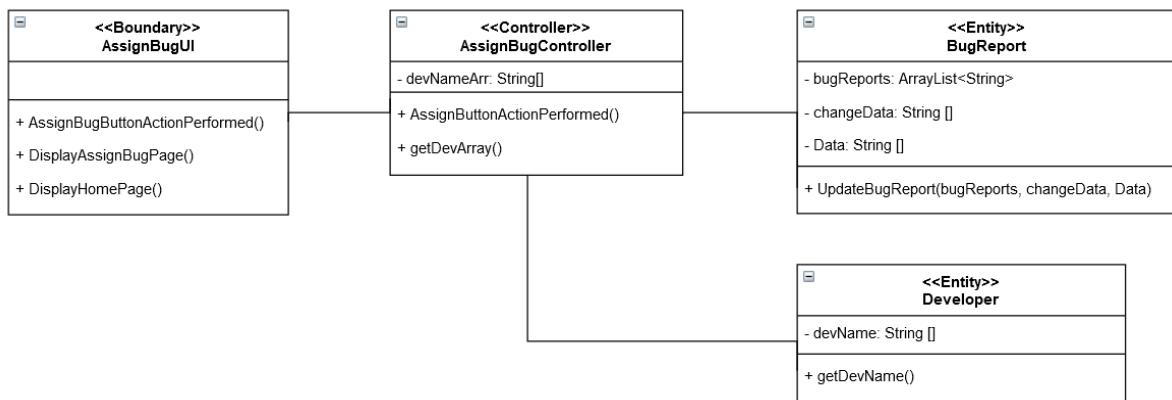
Use Case Diagram



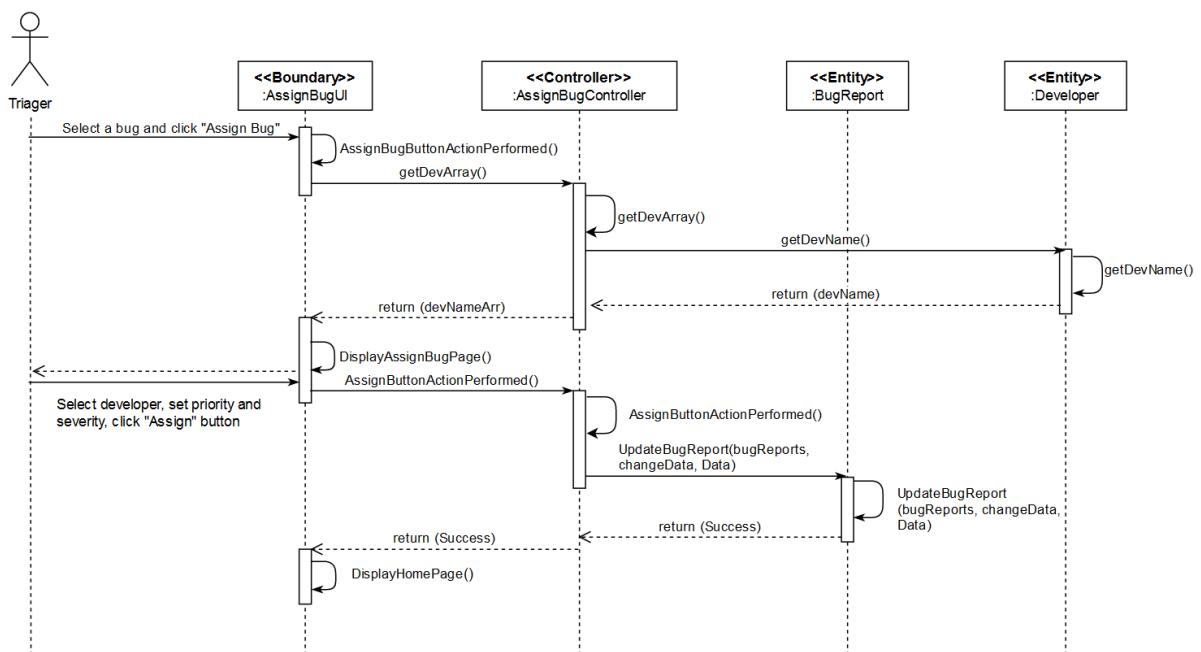
Use Case Description

Use Case Name: Assign Bugs	ID: 11
Stakeholders and goals: Triager want to assign bugs to the developer	
Description: There are bugs reported in the system required to be assigned to developer to resolve	
Actors: Triager	
Trigger: A triager selects a reported bug and click on the “Assign” button. The triager will then select from the drop-down list of developer name in the system, set the priority and severity of the bug and click on the “Assign” button	
Preconditions: <ol style="list-style-type: none">1. Triager must be logged in to the system2. There must be bug reported in the system	
Normal flow: <ol style="list-style-type: none">1. System displays a list of reported bugs at the home page2. Triager selects a bug he / she wants to assign3. System will direct triager to the assign page4. The triager will select from the drop-down list of developer name in the system, set the priority and severity of the bug and click on the “Assign” button5. System display the updated Assignee in the dashboard (homepage)6. End	
Sub flows: 4a. Traiger can select Low, Medium, High from the priority and severity drop down list	
Alternatives/Exceptional flows: None	

BCE Class Diagram



BCE Sequence Diagram



Wireframe

The wireframe displays the user interface for the 'Assign Bug' page:

- User's name:** Text input field.
- Home:** Button.
- Report Bug:** Button.
- Assign Bug:** Main section containing:
 - Bug ID:** Text input field.
 - Reporter:** Text input field.
 - Bug Title:** Text input field.
 - Date Reported:** Text input field.
 - Developer's List:** Text input field.
 - Current Status:** Text input field.
 - Set priority:** Text input field.
 - Set Severity:** Text input field.
 - Assign:** Button.

GUI

The image shows two screenshots of a bug tracking application interface.

Dashboard Screen:

- Welcome:** Barry White
- Navigation:** Home, Report Bug, View Tested Bugs
- Search:** Search here... (Bug ID dropdown, magnifying glass icon)
- Bug List:**

Bug ID	Title	Keyword	Date	Description
10000	Save Button not Working	Save Button	02/11/2020	Save Button not Working Save Button no
10001	Exit button not working	Exit button	02/11/2020	Exit button not working Exit button not wo
10002	Back Button not working	Back Button	02/11/2020	Back Button not working Back Button no
- Annotations:**
 - (1): Select a bug (points to the bug list area)
 - (2): Click assign bug (points to the Assign Bug button)

Assign Bug Screen:

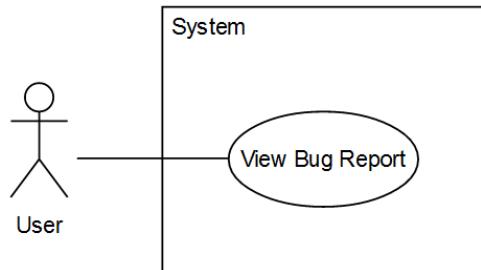
- Welcome:** Barry White
- Navigation:** Home, Report Bug, View Tested Bugs
- Form Fields:**

Bug ID: 10002	Reporter: Walter Green - U12345
Bug Title: Back Button not working	Date Reported: 02/11/2020
Developer's List:	
Developer's List:	Set priority:
(1): Select a developer (points to the developer dropdown menu)	Set priority (null, null, Low, Medium, High): (2): Choose the status (Not Resolved, Not Resolved, Pending, Resolved):
Set Severity:	Set Severity:
null	null
(3): Set the priority of the bug (points to the priority dropdown menu)	(4): Set the severity of the bug (points to the severity dropdown menu)
- Buttons:**
 - Assign (highlighted in blue)
- Annotations:**
 - (1): Select a developer (points to the developer dropdown menu)
 - (2): Choose the status (points to the status dropdown menu)
 - (3): Set the priority of the bug (points to the priority dropdown menu)
 - (4): Set the severity of the bug (points to the severity dropdown menu)
 - (5): Click assign to finish the assignment (points to the Assign button)

5.1.3 View Bug Report (User Story #241)

As a user, I want to be able to view the bug report of a bug being reported so that I can learn more about the bug.

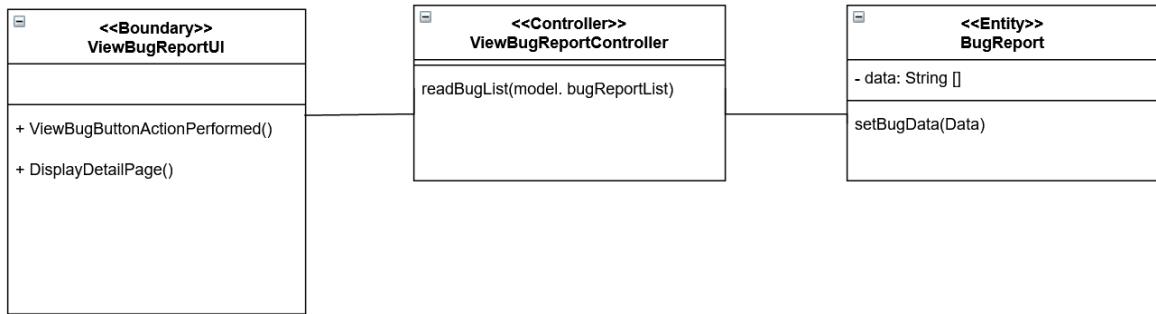
Use Case Diagram



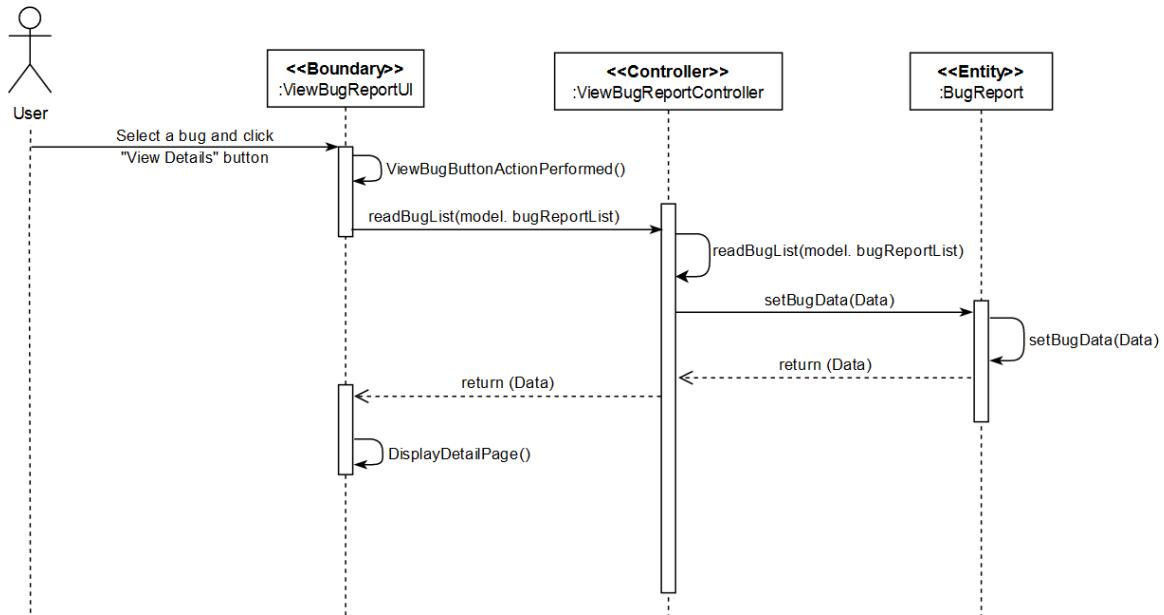
Use Case Description

Use Case Name: View Bug Report	ID: 241
Stakeholders and goals: User wants to view details of a bug report	
Description: Every reported bug should contains a bug report for user to view	
Actors: User	
Trigger: A user will select a reported bug displayed in the dashboard and click on the “View Details” button	
Preconditions: <ol style="list-style-type: none">1. User must be logged in to the system2. There must be bug reported in the system	
Normal flow: <ol style="list-style-type: none">1. System displays a list of reported bugs at the home page2. User selects a bug he / she wants to view and click “View Details” button3. System will direct user to the “Bug Specifications” page which shows all the details about the bug4. End	
Sub flows: None	
Alternatives/Exceptional flows: None	

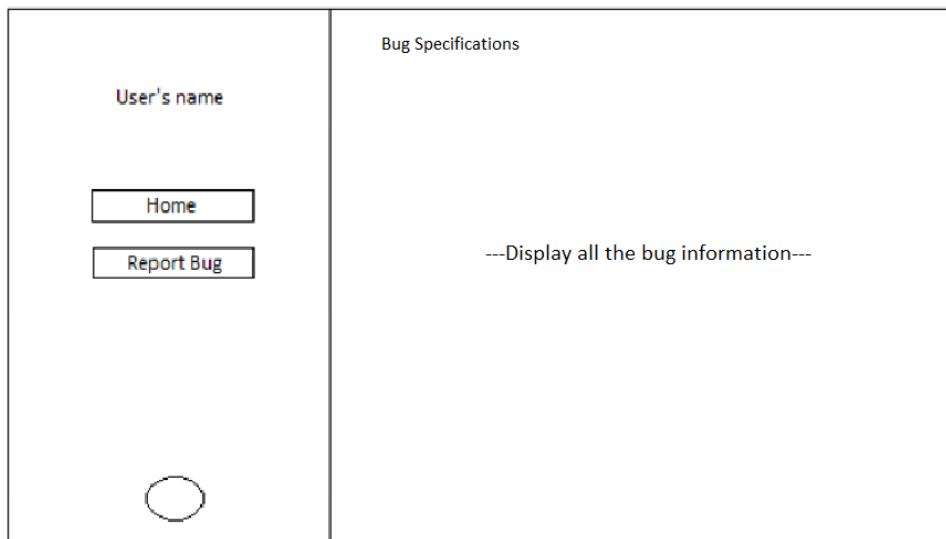
BCE Class Diagram



Sequence Diagram



Wireframe



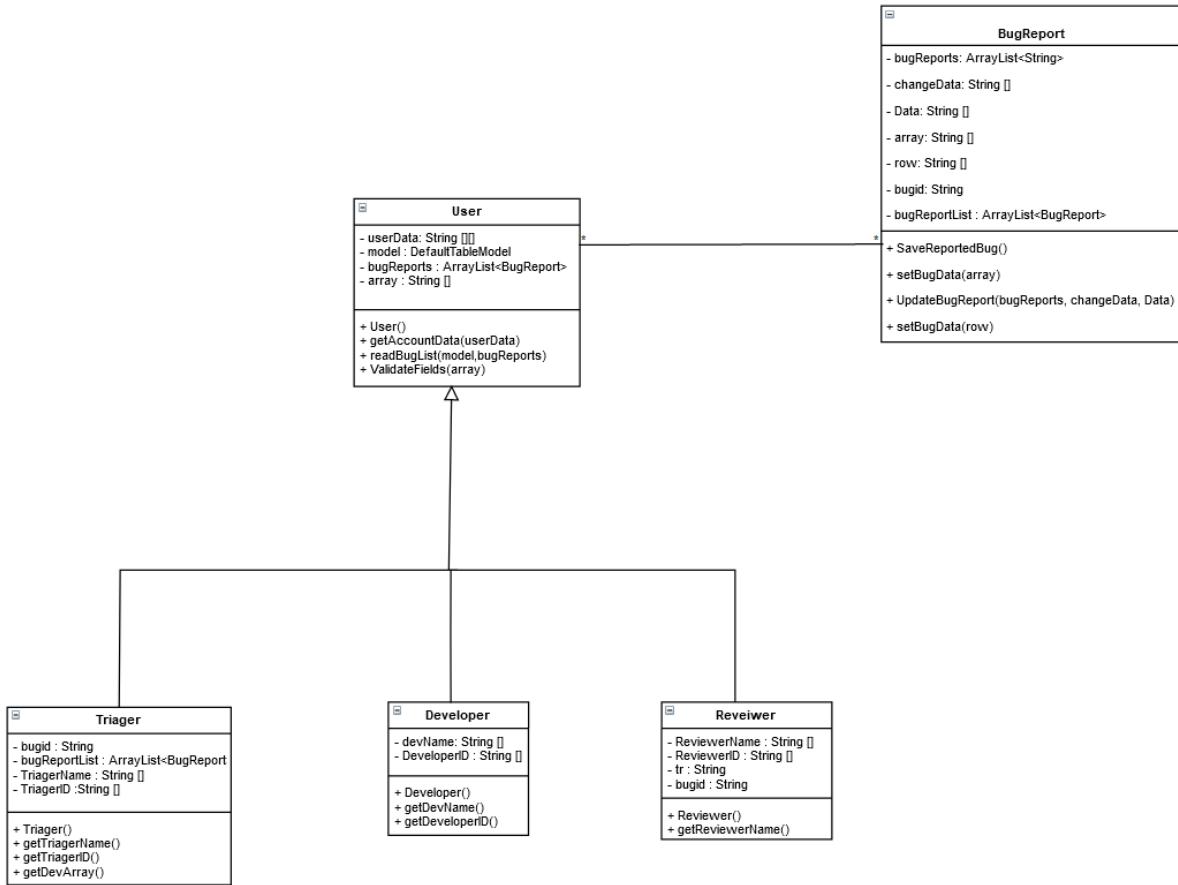
GUI

The screenshot displays a user interface for a bug tracking system. On the left, a dark blue sidebar labeled "Welcome" shows the user "Barry White". It has three buttons: "Home", "Report Bug", and "View Tested Bugs". Below these is a circular arrow icon. The main area is titled "Bug Specifications" and contains the following fields:

Field	Value
Bug ID	10002
Bug Title	Back Button not working
Reporter	Walter Green - U12345
Assignee	Barry White - T12345
Keyword	Back Button
Date Reported	02/11/2020
Developer	David White - D12345
Current Status	Pending
Severity	Medium
Priority	Medium
Submitted Patch	null
Reviewer	null

Below the main form, there are two scrollable text areas: "Bug Description" containing multiple lines of text about a back button issue, and "Test Results" containing the word "null".

5.2 Class Diagram



5.3 Test Cases

5.3.1 Functional Test Case

Report Bug

Project Name:	Bug Tracking System					
Test Case ID:	USER-01					
Test Case Description:	Check report bug function					
Created By:	Jun Wei					
Tested By:	Jun Wei					
Tested On:	10/10/2020					
Preconditions:	User must be logged in to the system					
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks
USER-01-4-1	Unsuccessful reporting - Empty cell in Bug Title	(1) Input "test" in all field except Bug Title (2) Click on Submit Bug button	A pop up message would appear warning user that the fields cannot be empty	Login Unsuccessful message dialog appears on screen saying "Fields cannot be empty"	Pass	Dialog box requires you to click ok
USER-01-4-2	Unsuccessful reporting - Empty cell in Issue Description	(1) Input "test" in all field except Issue Description (2) Click on Submit Bug button	A pop up message would appear warning user that the fields cannot be empty	Login Unsuccessful message dialog appears on screen saying "Fields cannot be empty"	Pass	Dialog box requires you to click ok
USER-01-4-3	Unsuccessful reporting - Empty cell in Bug Summary	(1) Input "test" in all field except Bug Summary (2) Click on Submit Bug button	A pop up message would appear warning user that the fields cannot be empty	Login Unsuccessful message dialog appears on screen saying "Fields cannot be empty"	Pass	Dialog box requires you to click ok
USER-01-4-4	Unsuccessful reporting - Empty cell in Keyword	(1) Input "test" in all field except Keyword (2) Click on Submit Bug button	A pop up message would appear warning user that the fields cannot be empty	Login Unsuccessful message dialog appears on screen saying "Fields cannot be empty"	Pass	Dialog box requires you to click ok
USER-01-4-5	Successful reporting - Normal User	(1) Input a valid Bug Title (2) Input a valid Issue Description (3) Input a valid Bug Summary (4) Input a valid Keyword (5) Click on Submit Bug Button	A pop up message would appear saying bug is reported	User dashboard page is displayed on the screen.	Pass	Dialog box requires you to click ok
USER-01-4-6	Successful reporting - Traiger	(1) Input a valid Bug Title (2) Input a valid Issue Description (3) Input a valid Bug Summary (4) Input a valid Keyword (5) Click on Submit Bug Button	A pop up message would appear saying bug is reported	Traiger dashboard page is displayed on the screen.	Pass	Dialog box requires you to click ok
USER-01-4-7	Successful reporting - Developer	(1) Input a valid Bug Title (2) Input a valid Issue Description (3) Input a valid Bug Summary (4) Input a valid Keyword (5) Click on Submit Bug Button	A pop up message would appear saying bug is reported	Developer dashboard page is displayed on the screen.	Pass	Dialog box requires you to click ok
USER-01-4-8	Successful reporting - Reviewer	(1) Input a valid Bug Title (2) Input a valid Issue Description (3) Input a valid Bug Summary (4) Input a valid Keyword (5) Click on Submit Bug Button	A pop up message would appear saying bug is reported	Reviewer dashboard page is displayed on the screen.	Pass	Dialog box requires you to click ok

Assign Bug

Project Name:	Bug Tracking System							
Test Case ID:	TRIAGER-01							
Test Case Description:	Assign bug function							
Created By:	Jun Wei							
Tested By:	Jun Wei							
Tested On:	10/10/2020							
Preconditions:	Triager must be logged in							
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail	
TRIAGER-01-1	Successful assign to developer - Triager	(1) Click on a reported bug (2) Click on Assign Bug button (3) Select valid option from dropdown menu in Developer's List (4) Select valid option from dropdown menu in Current Status (5) Select valid option from dropdown menu in set priority (6) Select valid option from dropdown menu in set Severity (7) Click on Assign button	Assign successful message dialog appears on screen saying "Assign Bug Successful"	Triager dashboard page is displayed on the screen.	Pass			

Set Severity of Bug (Inside Assign Bug)

Project Name:	Bug Tracking System						
Test Case ID:	TRIAGER-01						
Test Case Description:	Fix bug severity function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	10/10/2020						
Preconditions:	Triager must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
TRIAGER-01-2	Successfully assigning the severity of the bug - Triager	(1) Select severity through drop down menu. (2) Click on Assign button.	Assign successful. The system would inform the developer	Triager dashboard page is displayed on the screen.	Pass		

Set Priority of Bug (Inside Assign Bug)

Project Name:	Bug Tracking System						
Test Case ID:	TRIAGER-01						
Test Case Description:	Fix bug priority function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	10/10/2020						
Preconditions:	Triager must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
TRIAGER-01-3	Successfully setting the Priority of the bug - Triager	(1) Select priority through drop down menu. (2) Click on Assign button.	Assign successful. The system would inform the developer	Triager dashboard page is displayed on the screen.	Pass		

View Bug Report

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	View bug report function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	10/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1-1	Successful view the specifications of a bug - View by all user	(1) Click on a reported bug (2) Click on "View Details" button	System will open the selected bug report and display all the specifications	"Bug specifications" page is displayed	Pass		

5.4 Evidence of Use of Methodologies

5.4.1 Meeting Minute

Project Name: Bug tracking system

Meeting Objection: Prepare and begin sprint 2

Date: 8th October 2020

Attendee

Ng Ming Yao

Lim Li Shan

Neo Zhi Kai

Liu Zihan

Mallah Rahul Premchand

Chong Jun Wei

Ivan Lee You Qing

Agenda:

- Define what can be delivered in sprint 2
- Decide on the goal for Sprint 2
- Revise on Product Backlog
- Work on Sprint 2 Backlog

S/No	Item	Action Item	Due Date
1	Distribution of tasks	Task distributed by MingYao	8 th October 2020
2	Add/Remove user stories to backlog	To be done by team	8 th October 2020
3	Create sprint 2 backlog	Done by Zhi Kai	8 th October 2020
4	Complete all tasks in Sprint 2	Done by team	14 th October 2020

Date and time of next meeting:

Sprint 3: 15th October 2020

Meeting minutes 2 prepared by Zhi Kai

5.4.2 Sprint Planning Meeting

Taiga - Product Backlog (Edited Items)

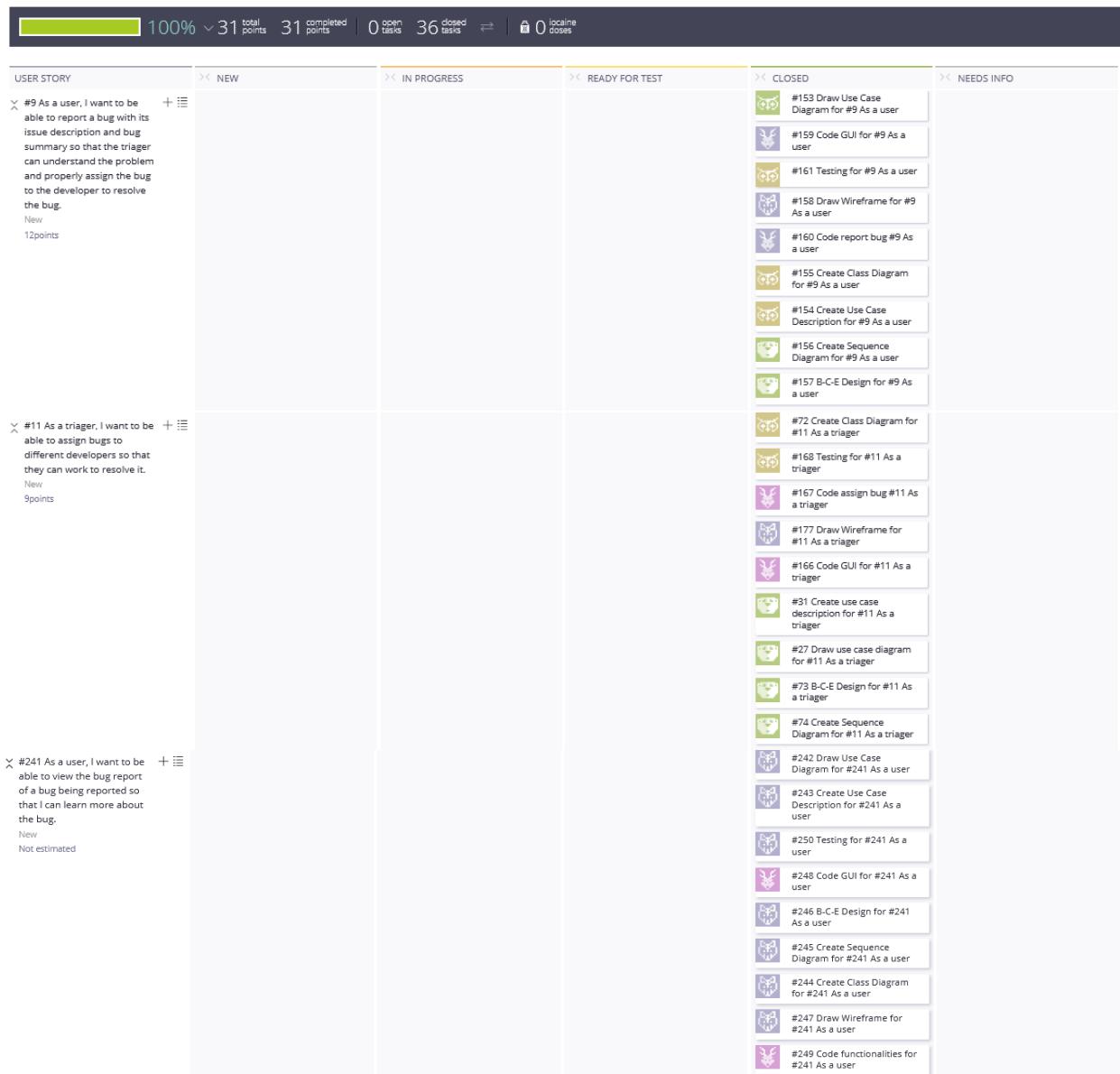
BACKLOG SIM2020Q4 - TEAM7268

The screenshot shows the Taiga Product Backlog interface for the SIM2020Q4 team. At the top, there's a progress bar indicating 9% completion, 136 defined points, 12 closed points, and 12 points per sprint. Below the header, there are filters for SHOW FILTERS, SHOW TAGS, and VELOCITY FORECASTING, along with a button to '+ ADD A NEW USER STORY' and a grid icon.

Votes	User Stories	Status	Points
<input type="checkbox"/>	▲ 0 #18 As a developer, I want to be able to check for my assigned task's status and problem reported so that I can begin fixing it.	New ▾	12
<input type="checkbox"/>	▲ 0 #19 As a developer, I want to be able to submit my assigned task so that it can be tested by the reviewers.	New ▾	9
<input type="checkbox"/>	▲ 0 #22 As a reviewer, I want to be able to input my test result so that triager could mark the bug as solved.	New ▾	7
<input type="checkbox"/>	▲ 0 #13 As a triager, I want to be able to set a bug as invalid so that invalid bugs can be ignored.	New ▾	6
<input type="checkbox"/>	▲ 0 #14 As a triager, I want to be able to set the status of a bug as solved so that it can show that the bug has been resolved.	New ▾	7
<input type="checkbox"/>	▲ 0 #3 4. As a user, I want to be able to search a bug through keywords so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6
<input type="checkbox"/>	▲ 0 #6 5. As a user, I want to be able to search a bug through titles so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6
<input type="checkbox"/>	▲ 0 #4 6. As a user, I want to be able to search a bug through assignee so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6
<input type="checkbox"/>	▲ 0 #5 7. As a user, I want to be able to search a bug through reporter name so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6
<input type="checkbox"/>	▲ 0 #2 8. As a user, I want to be able to search a bug through bug ID so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6
<input type="checkbox"/>	▲ 0 #15 6. As a triager, I want to be able to generate a report that shows the total numbers of bugs reported in a selected timeline with all the information for each bug so that I can see the performance of reporters.	New ▾	7
<input type="checkbox"/>	▲ 0 #16 7. As a triager, I want to be able to generate a report that shows the total numbers of bugs resolved in a selected timeline with all information for each bug so that I can see the performance of developers.	New ▾	7
<input type="checkbox"/>	▲ 0 #8 As a user, I want to be able to provide comments so that I can participate in the discussion of a certain bug.	New ▾	8

5.4.3 Taiga (End of Sprint)

SPRINT 2 SIM2020Q4 - TEA... 08 OCT 2020-14 OCT 2020



5.4.4 Sprint Review Meeting

Work accomplished during Sprint 2:

- Revise and enhance existing product backlog items
- Team decided on the goal for sprint 2
- Completed all tasks for items selected from product backlog to sprint 2

5.4.5 Sprint Retrospective Meeting

Work to complete next reporting period:

- Preparation for Sprint 3
- Reprioritize features according to requirements

What went well in the Sprint:

- Team members are able to agree and accept on a similar goal thus making it easier for the project to be carried out
- Team members are good in their respective tasks as they are assigned according to their strength and weakness and therefore the assigned tasks in the sprint is able to proceed smoothly
- Goals and instructions for each individual are clear, team members are aware of the direction we are moving

What can be improved:

- Focus on the initial goal and not go off track

(Discussions) Suggestions/Issues:

- Have to carefully review the priorities of the user stories and the goals we set so to ensure we can complete the required tasks in a sprint on schedule
- To use a common software for the diagrams so teammates can make amendments more easily

5.4.6 Member Contribution

Name:	Task Completed	Contribution
Ng Ming Yao	<ul style="list-style-type: none"> • Distribution of roles • Update class diagrams for sprint 2 user stories • Update B-C-E diagrams for sprint 2 user stories • Update sequence diagrams for sprint 2 user stories • Code GUI for sprint 2 user stories • Code functionalities for sprint 2 user stories 	
Lim Li Shan	<ul style="list-style-type: none"> • Update B-C-E diagrams for sprint 2 user stories • Update sequence diagrams for sprint 2 user stories • Create week 2 report 	
Neo Zhi Kai	<ul style="list-style-type: none"> • Create meeting minutes 2 • Create week 2 report • Create use case diagrams for sprint 2 user stories • Create use case descriptions for sprint 2 user stories • Update class diagrams for sprint 2 user stories • Create B-C-E diagrams for sprint 2 user stories • Create sequence diagrams for sprint 2 user stories • Analysed and evaluated product backlog • Prepare sprint 2 backlog 	
Liu Zihan	<ul style="list-style-type: none"> • Create use case diagrams for sprint 2 user stories • Create use case descriptions for sprint 2 user stories • Create B-C-E diagrams for sprint 2 user stories • Create sequence diagrams for sprint 2 user stories 	
Mallah Rahul Premchand	<ul style="list-style-type: none"> • Update class diagrams for sprint 2 user stories • Update B-C-E diagrams for sprint 2 user stories • Update sequence diagrams for sprint 2 user stories • Code GUI for sprint 2 user stories • Code functionalities for sprint 2 user stories 	
Chong Jun Wei	<ul style="list-style-type: none"> • Create and complete testing for sprint 2 user stories • Create class diagram draft for all entities 	
Ivan Lee You Qing	<ul style="list-style-type: none"> • Create class diagram draft for all entities • Create first batch of bug reports 	

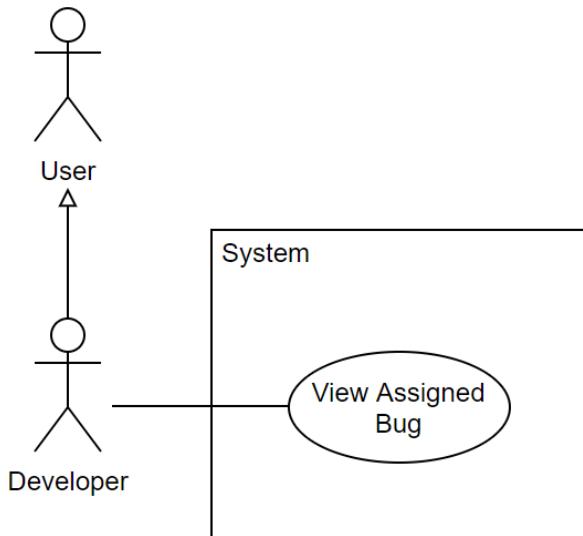
6 Sprint 3

6.1 User Stories and Diagrams

6.1.1 View Assigned Bug (User Story #18)

As a developer, I want to be able to check for my assigned task's status and problem reported so that I can begin fixing it.

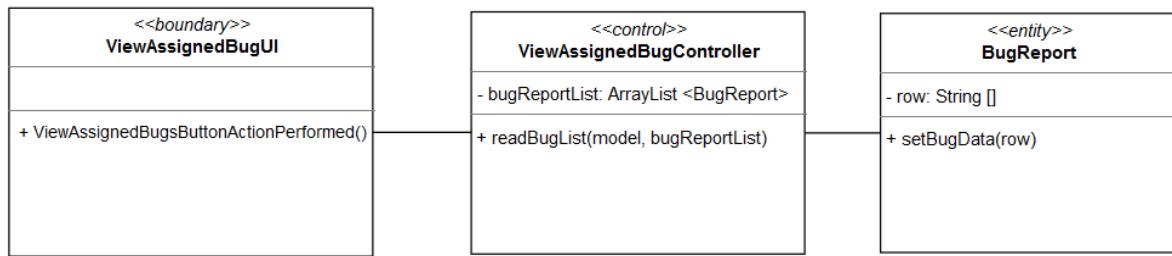
Use Case Diagram



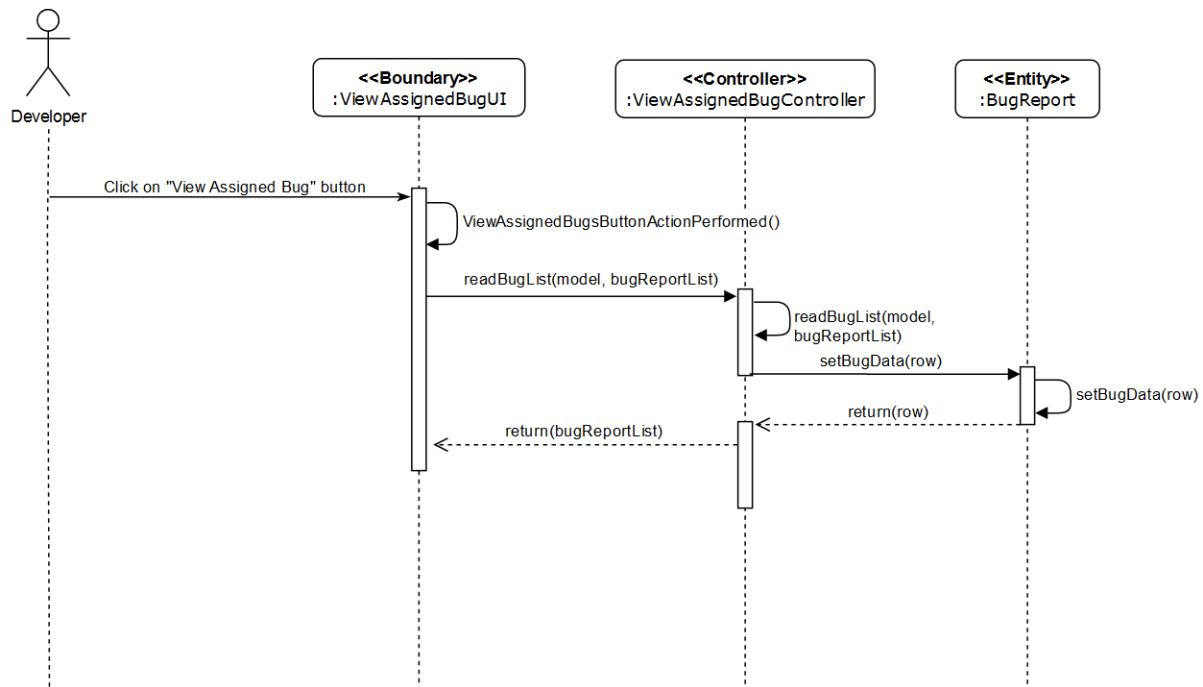
Use Case Description

Name: View Assigned Bug	ID: 18
Stakeholders and goals: Developer - wants to view the bugs that was assigned to them by Triager	
Description: Developer is able to view all the bugs that are assigned to him/her in the dashboard	
Actors: Developer	
Pre-conditions: Developer must log in to the system using his/her username and password	
Trigger: Developer clicks on "View Assigned Bugs" button	
Normal Flow: <ol style="list-style-type: none">1. The system displays all the bugs reported in the dashboard (Homepage)2. Developer clicks on "View Assigned Bugs" button3. The system now displays all the bugs assigned to the developer in the dashboard4. End	
Sub-flows: None	
Alternative/Exceptional flows: 3: Empty Data: There is currently no assigned bug to this developer	

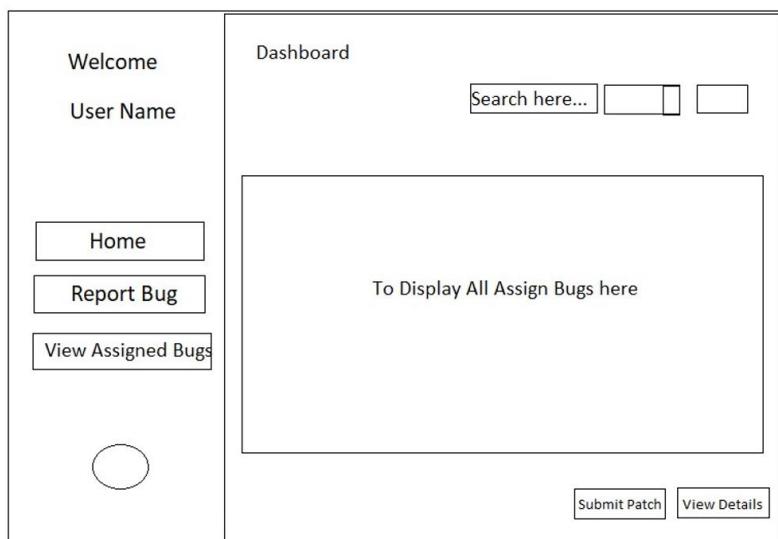
BCE Class Diagram



Sequence Diagram



Wireframe



GUI

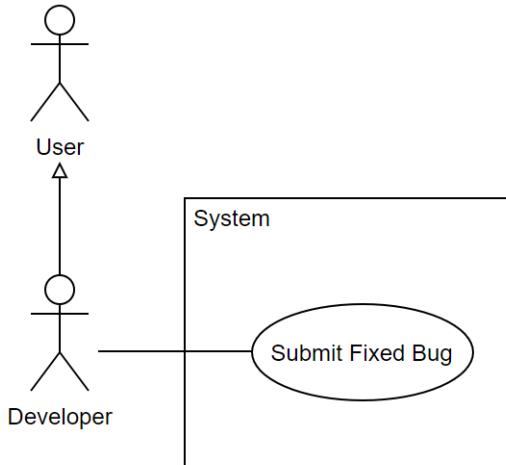
The screenshot shows a software application window titled "Dashboard". The left sidebar is dark blue with white text, displaying "Welcome" and "David White" at the top, followed by three buttons: "Home", "Report Bug", and "View Assigned Bugs". Below these buttons is a circular icon with a downward arrow. The main area is light blue and contains a search bar with placeholder text "Search here..." and a dropdown menu labeled "Bug ID" with a magnifying glass icon. A table lists 10 bugs, each with columns for Bug ID, Bug Title, Keyword, Date, and Description. At the bottom right of the main area are two buttons: "Submit Patch" and "View Details".

Bug ID	Bug Title	Keyword	Date	Description
10002	Username login caps sen...	Login	11/10/2020	User must type according to caps sen
10004	System Crash when clicki...	Logout	11/10/2020	System crash when logging out
10007	Comment edited but still ...	Edit	11/10/2020	Edited comment is not shown
10010	Non-ASCII characters ge...	Search	12/10/2020	Error cause of non-ascii characters
10023	Error in generating the re...	Report bug	13/10/2020	Error in generating the correct date
10029	Error in showing the corre...	Search bug	13/10/2020	Error in the search engine.
10047	Logging out takes a log fi...	Logout	15/10/2020	Long waiting time while logging out
10050	Developer does not get n...	Notification	16/10/2020	Missing notifications
10051	Developer are not able t...	Comment	16/10/2020	Developer are not able to see the sug
10055	Search text did not appe...	Search	16/10/2020	Search text missing in search box
10069	Reviewer are not able to ...	23rd October ...	17/10/2020	UI2371
10073	"More results" doesn't work	Search	18/10/2020	"More results" produce nothing

6.1.2 Submit Fixed Bug (User Story #19)

As a developer, I want to be able to submit my assigned task so that it can be tested by the reviewers.

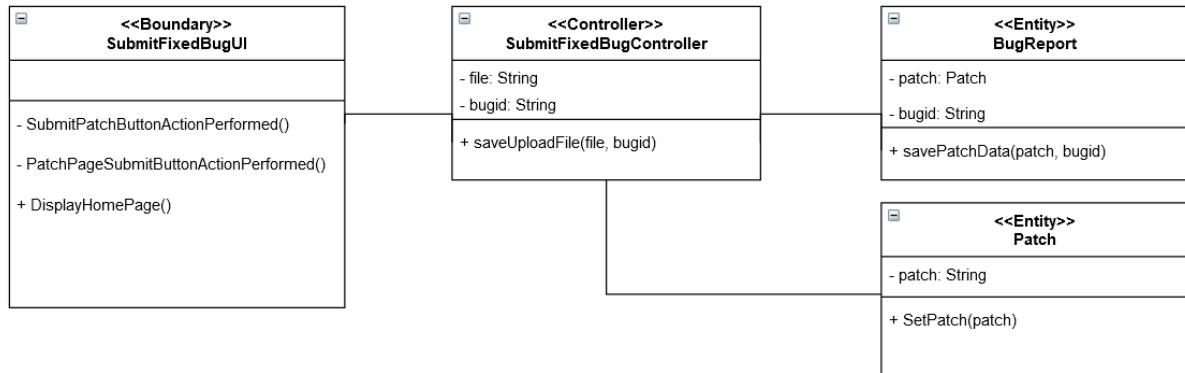
Use Case Diagram



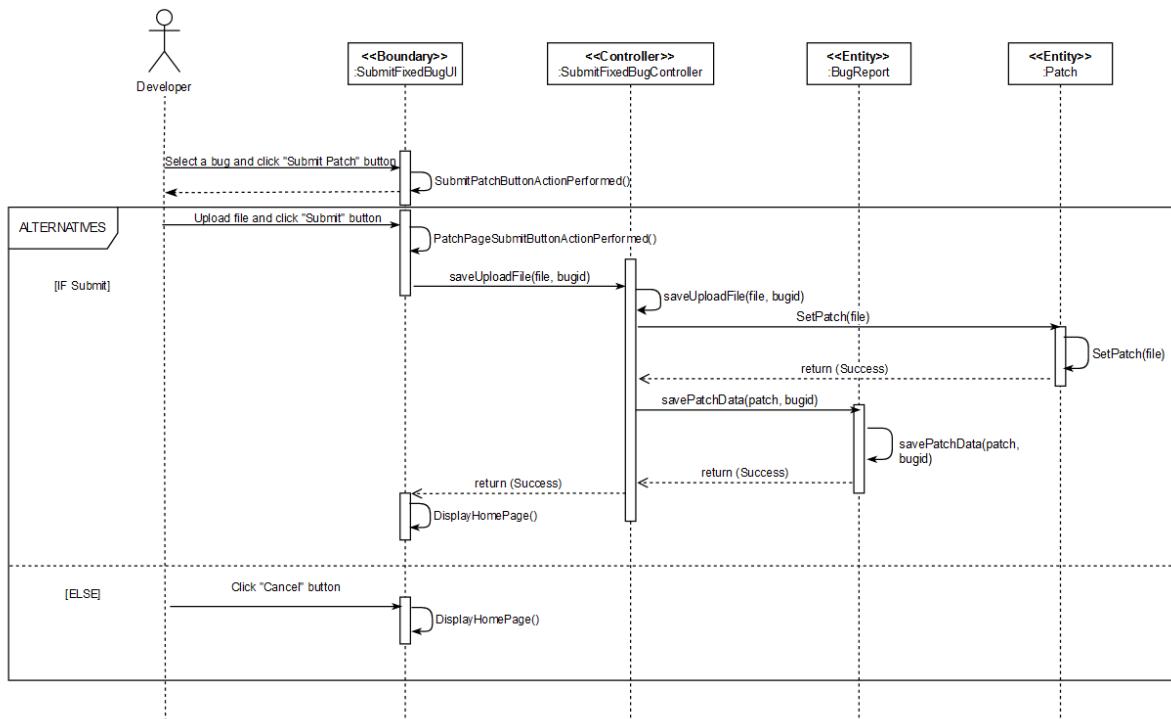
Use Case Description

Name: Submit Fixed Bug	ID: 19
Stakeholders and goals: Developer - wants to submit the patches for bugs that he has fixed to the reviewer	
Description: Developer click patch submission button to submit the patches for bugs that are assigned to him	
Actors: Developer	
Pre-conditions: Developer must log in to the system using his/her username and password	
Trigger: Developer selects any reported bug and click the “Submit Patch” button. In the patches submission page, developer can attach files as resolution and click “Submit”.	
Normal Flow: <ol style="list-style-type: none">1. The system displays the developer's home screen2. Developer selects any reported bug and click the “Submit Patch” button3. System will direct developer to the Patches Submission page4. The Patches Submission page will display the Bug's ID, bug title, reporter name and date reported which are all non-editable fields and a field to attach file5. Developer clicks on the “Attach” button to upload the patches/results he has completed6. Developer clicks on the “Submit” button to save7. End	
Sub-flows: 4a. Submission Cancellation: Developer clicks on “Cancel” button to quit submitting work	
Alternative/Exceptional flows: None	

BCE Class Diagram



Sequence Diagram



Wireframe

The wireframe shows a two-column layout:

- Left Column (Welcome):**
 - User Name
 - Home
 - Report Bug
 - View Assigned Bugs
- Right Column (Patches Submission):**
 - Bug ID
 - Reporter
 - Bug Title
 - Date Reported
 - Attach File
 - Attach
 - [Cancel]
 - Submit

GUI

Patches Submission

Bug ID: 10000 Reporter: Walter Green - UI12345

Bug Title: Save Button not Working Date Reported: 02/11/2020

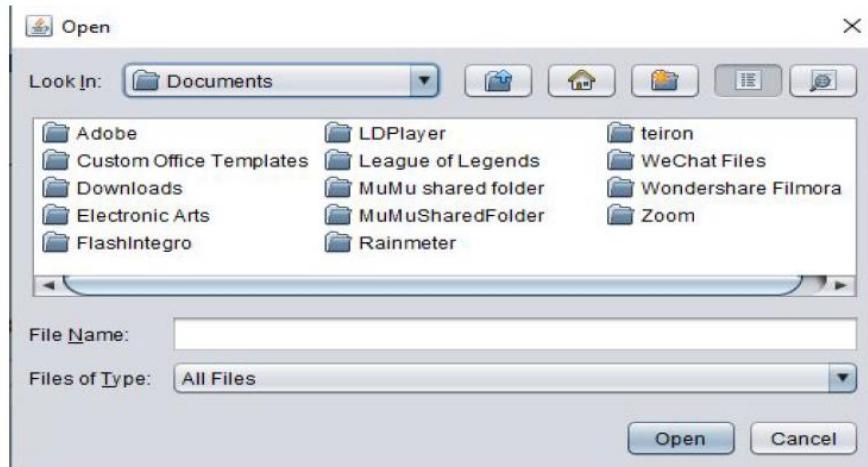
Attach File: elio.exe Attach

(1) Click here to submit the patch file

Cancel Submit (2) Press submit to submit the file and go back to previous page

Home Report Bug View Assigned Bugs

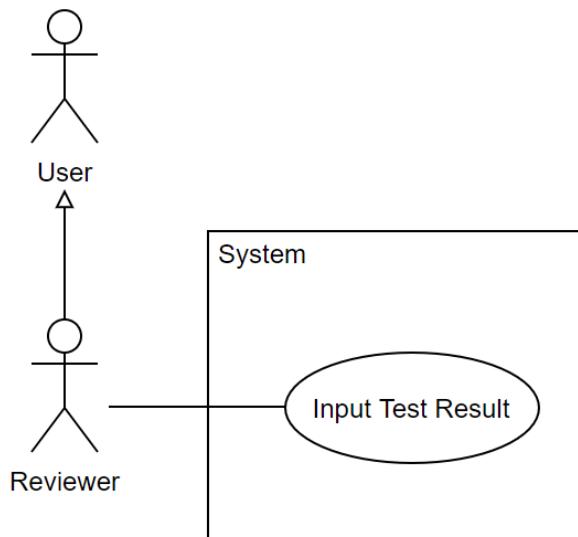
Welcome
David White



6.1.3 Input Test Result (User Story #22)

As a reviewer, I want to be able to input my test result for all bugs that have already been patched so that triager could mark the bug as solved.

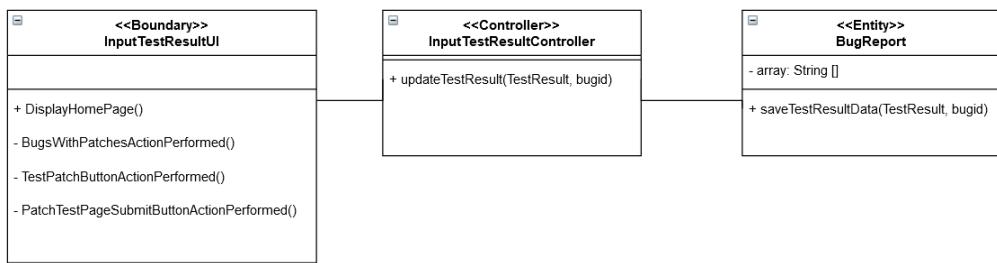
Use Case Diagram



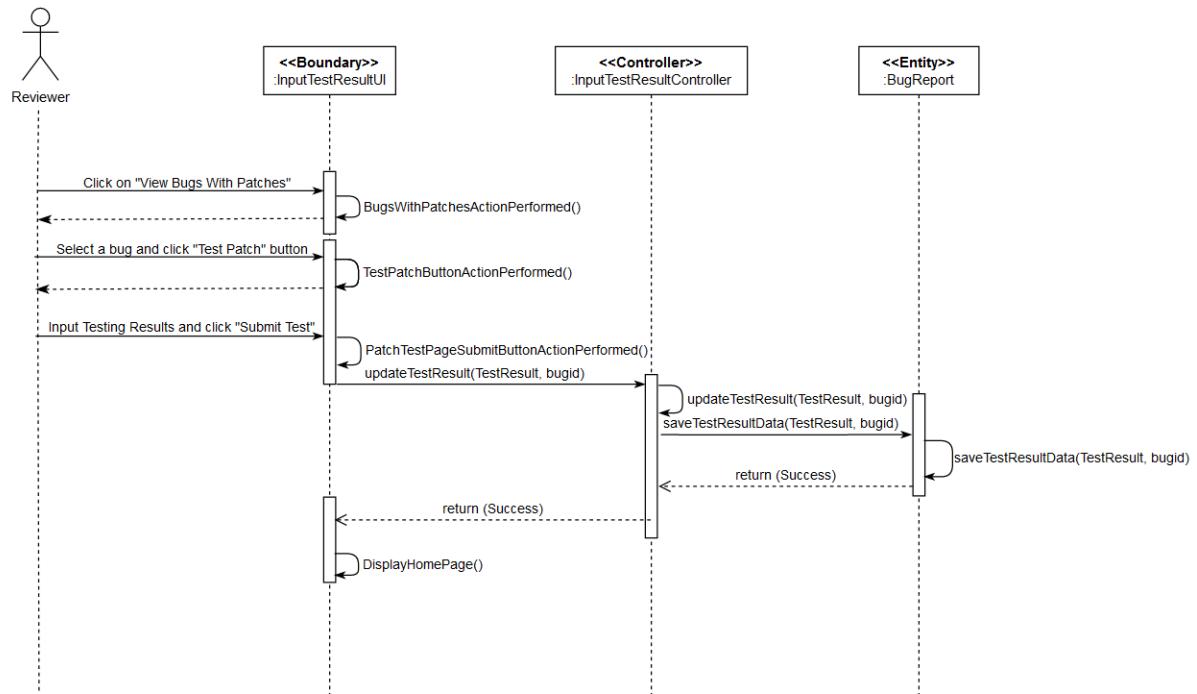
Use Case Description

Name: Input Test Result	ID: 22
Stakeholders and goals: Reviewer – want to input test result into the system	
Description: Reviewer test the bug using the patch provided and input the test result	
Actors: Reviewer	
Pre-conditions: Reviewer must log in to the system using his/her username and password	
Trigger: The reviewer clicks the “View Bugs With Patches” button and select a bug report and click “Test Patch” button	
Normal flow: <ol style="list-style-type: none">Reviewer selects a bug report and clicks “Test Patch” buttonSystem display the “Input Patch Test Results” screenInside “Input Patch Test Results” screen will display the bug’s ID, Title, Date Reported, Reporter, Developer and the Patch SubmittedReviewer input test result into the system and clicks on “Submit Test”The system stores the result and update the databaseThe system transfers the bug to the triagerEnd	
Sub-flows: None	
Alternative/Exceptional flows: <ol style="list-style-type: none">a. Empty Data: System contains no bug that has been patched	

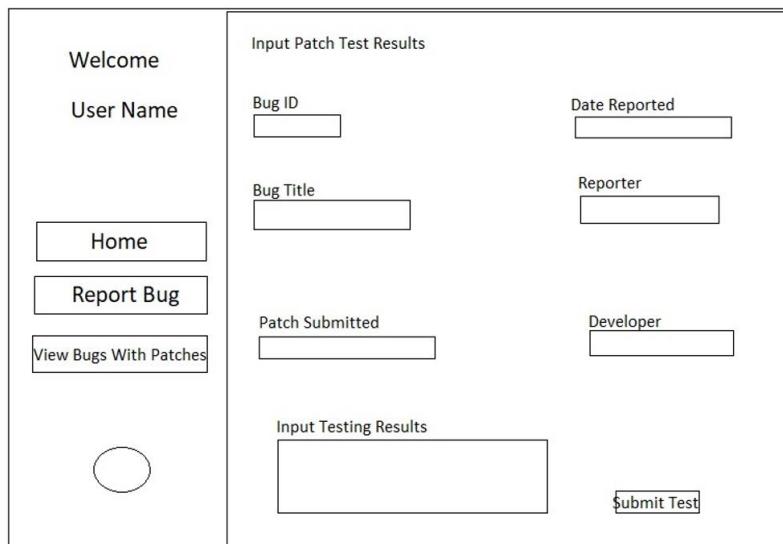
BCE Class Diagram



Sequence Diagram



Wireframe



GUI

The screenshot shows a software interface with a dark blue sidebar on the left and a light blue main content area.

Left Sidebar (Dark Blue):

- Welcome
- John Davis
- [Home](#)
- [Report Bug](#)
- [View Bugs With Patches](#)

Main Content Area (Light Blue):

Input Patch Test Results

Bug ID: 10000 Date Reported: 02/11/2020

Bug Title: Save Button not Working Reporter: Walter Green - U12345

Patch Submitted: ello.exe Developer: David White - D12345

Input Testing Results:

Patch successfully tested. The bug issue has been resolved.

Buttons:

- (1): Write the testing result here
- (2): Submit Test
- (2): Submit the testing result

Bug List Table:

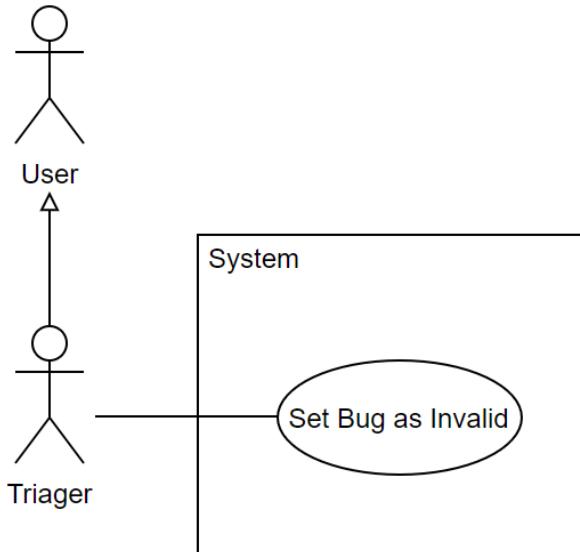
Developer	Current Status	Priority	Severity	Patch Su...	Reviewer	Test Results
David White - D12345	Pending	Low	Medium	ello.exe	John Da...	Patch successfully t...
David White - D12345	Pending	Medium	Medium	null	null	null
David White - D12345	Pending	Low	Medium	ello.exe	John Da...	Patch successfully t...
David White - D12345	Pending	Medium	Medium	null	null	null

In the bug list, a successful message will be shown

6.1.4 Set Bug as Invalid (User Story #13)

As a triager, I want to be able to set a bug as invalid so that invalid bugs can be ignored.

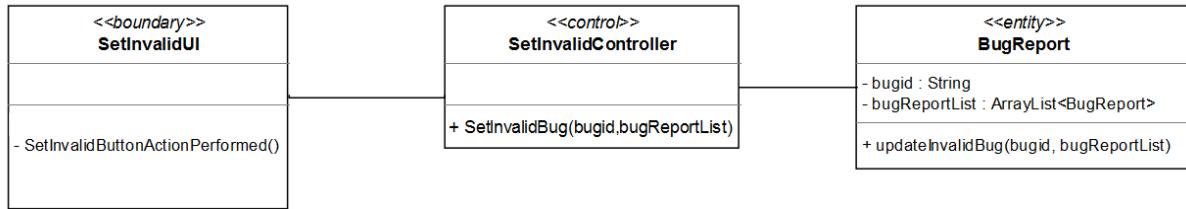
Use Case Diagram



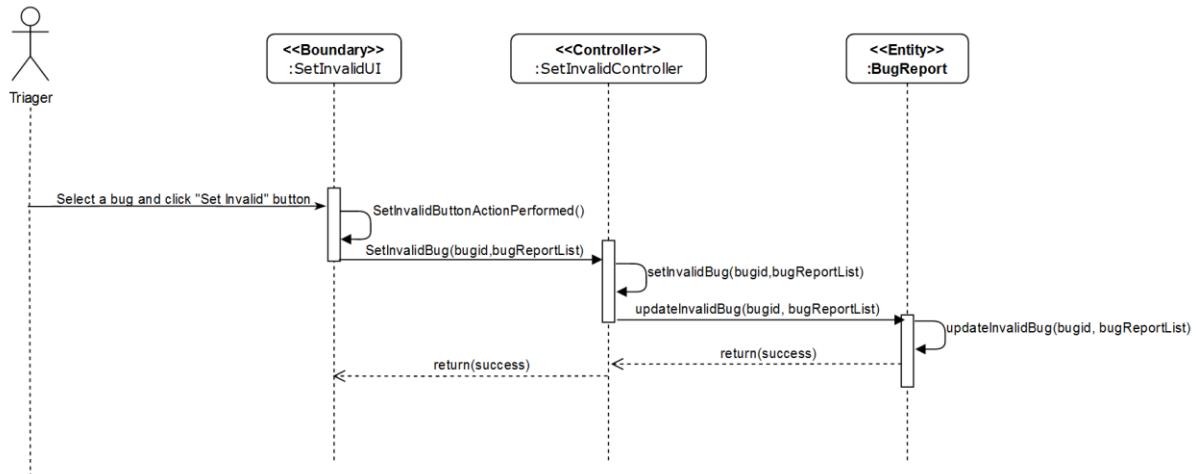
Use Case Description

Use Case Name: Set Bug as Invalid	ID: 13
Stakeholders and goals: Triager want to set a reported bug to invalid so that it can be ignored	
Description: A triager wants to set a status of a reported bug to invalid	
Actors: Traiger	
Trigger: A triager set a reported bug status to invalid	
Preconditions: <ol style="list-style-type: none">1. Triager must be logged in to the system2. There must be bug reported in the system	
Normal flow: <ol style="list-style-type: none">1. System displays a list of reported bugs at the home page2. Triager selects a bug report and click on the “Set Invalid” button3. System will change the current status of the bug report to “Invalid”4. End	
Sub flows: None	
Alternatives/Exceptional flows: None	

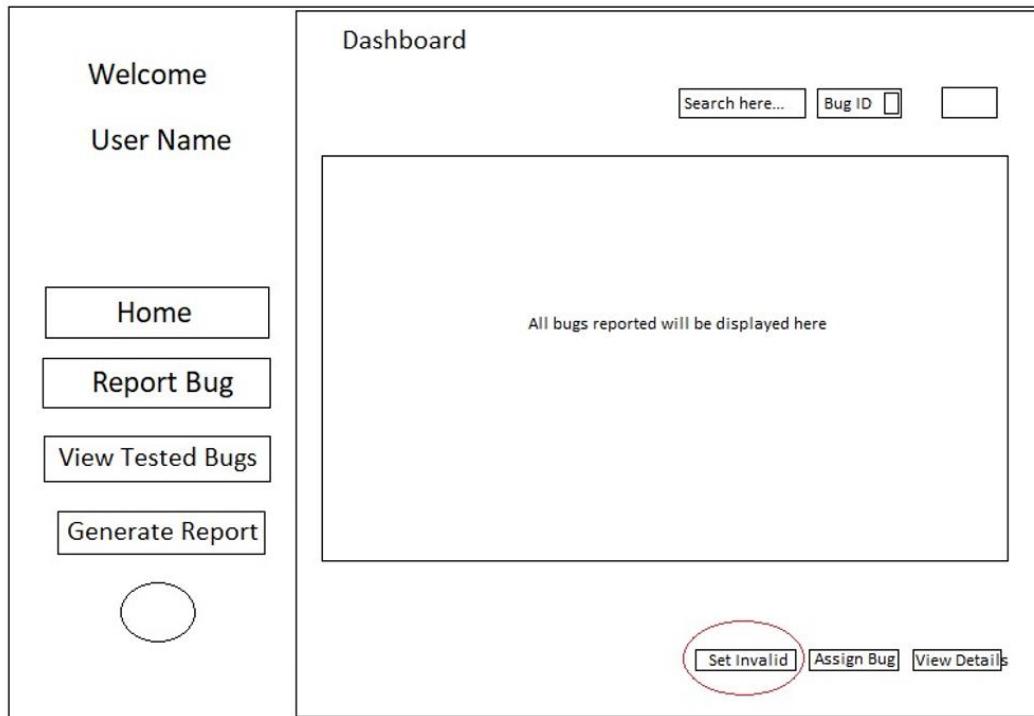
BCE Class Diagram



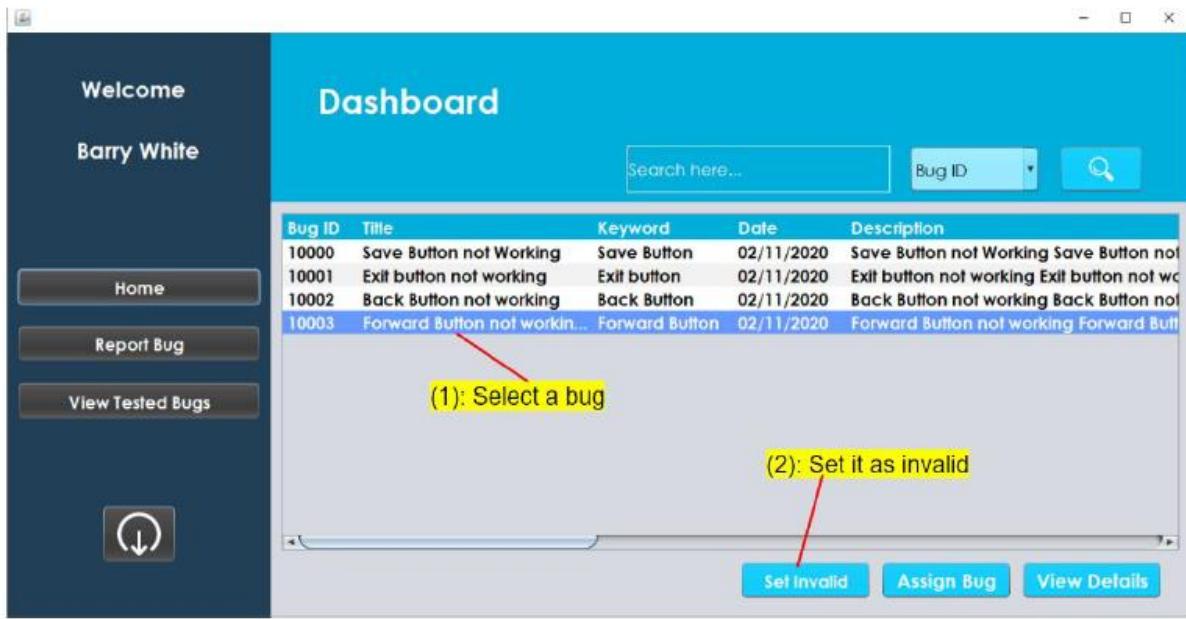
Sequence Diagram



Wireframe



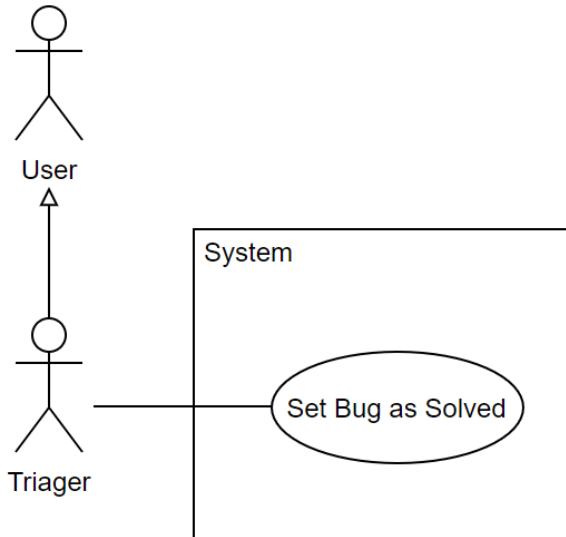
GUI



6.1.5 Set Bug as Solved (User Story #14)

As a triager, I want to be able to set the status of a bug as solved so that it can show that the bug has been resolved.

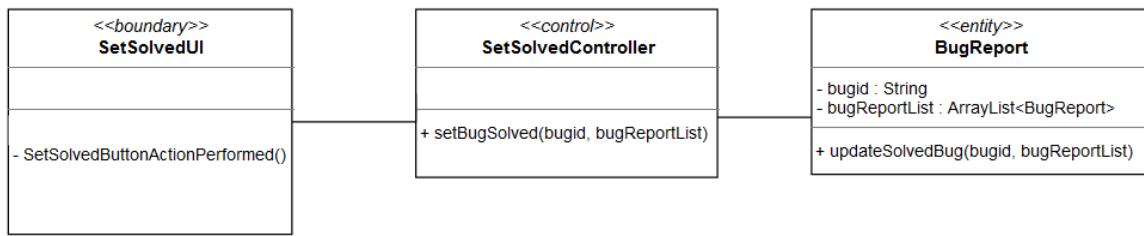
Use Case Diagram



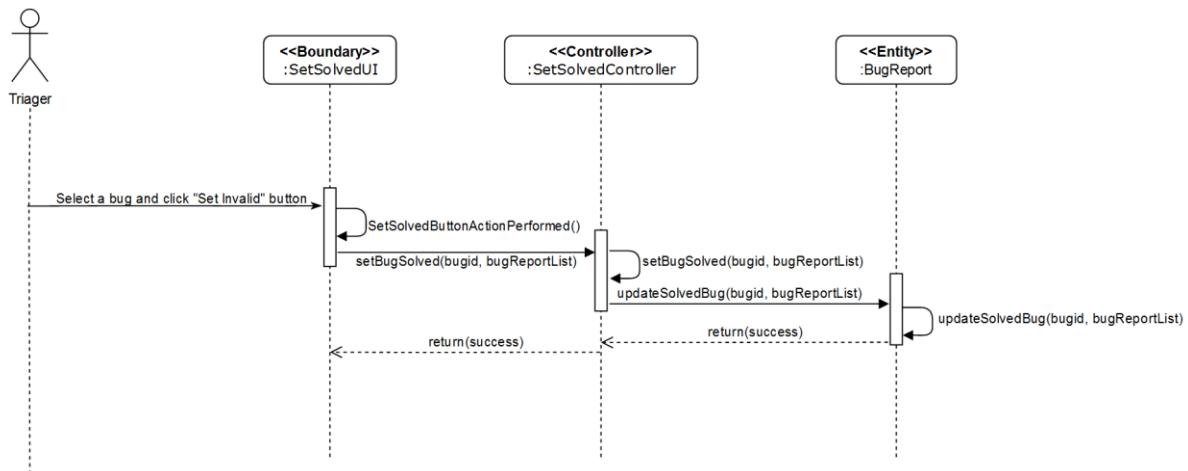
Use Case Description

Use Case Name: Set Bug as Solved	ID: 14
Stakeholders and goals: Triager wants to set a reported bug as solved	
Description: A triager wants to set a status of a reported bug to "Resolved"	
Actors: Triager	
Trigger: A triager in a system set a reported bug status to "Resolved"	
Preconditions: <ol style="list-style-type: none">1. Triager must be logged in to the system2. There must be bug reported in the system	
Normal flow: <ol style="list-style-type: none">1. System displays a list of reported bugs at the home page2. Triager selects a bug report and click on the "Set Bug Solved" button3. System will change the current status of the bug report to "Resolved"4. End	
Sub flows: None	
Alternatives/Exceptional flows: None	

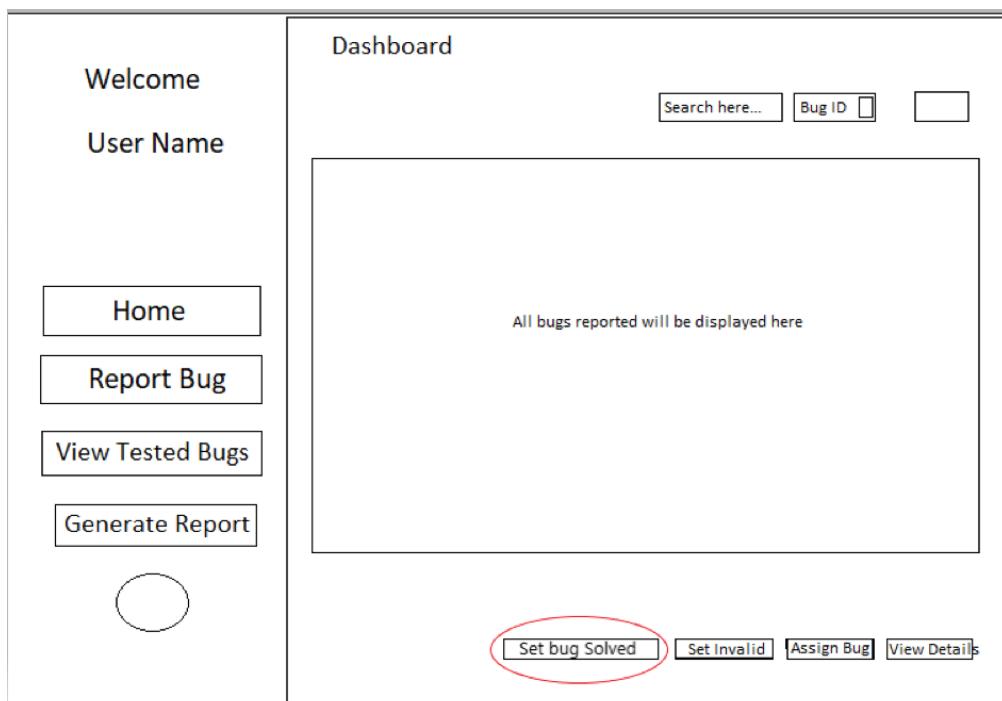
BCE Class Diagram



Sequence Diagram



Wireframe



GUI

The screenshot shows a desktop application window titled "Dashboard". On the left, there's a sidebar with a dark blue header containing "Welcome" and "Barry White". Below the header are four buttons: "Home", "Report Bug", "View Tested Bugs", and "Generate Report". At the bottom of the sidebar is a circular icon with a downward arrow. The main area has a light blue header with the title "Dashboard". Below the header are search fields for "Search here..." and "Bug ID" with a magnifying glass icon. A table lists 12 bugs, each with columns for Bug ID, Title, Keyword, Date, and Description. The first few rows of the table are:

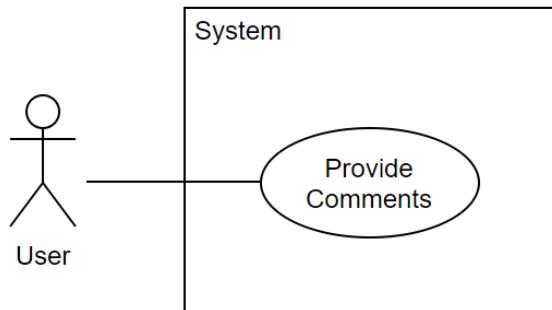
Bug ID	Title	Keyword	Date	Description
10000	Application crash on clic...	Save button	11/10/2020	System crash when clicking SAVE
10001	Unable to delete user	Delete button	11/10/2020	Admin cannot delete user
10002	Username login caps sen...	Login	11/10/2020	User must type according to caps sens
10003	New accounts cannot ch...	Password	11/10/2020	User cannot change password
10004	System Crash when clickl...	Logout	11/10/2020	System crash when logging out
10005	Logout dialog box remains	Logout	11/10/2020	Shows logout but dialog box stays
10006	Unable to edit comment	Edit	11/10/2020	User cannot edit comment
10007	Comment edited but still s...	Edit	11/10/2020	Edited comment is not shown
10008	No results found when se...	Search	11/10/2020	User cannot search
10009	Search results open in a n...	Search	11/10/2020	User search results in a new window
10010	Non-ASCII characters ge...	Search	12/10/2020	Error cause of non-ascii characters
10011	Advanced search functio...	Search	12/10/2020	Advanced search function not working
10012	Search results not refresh...	Search	12/10/2020	Search results not refresh after search

At the bottom of the table are four buttons: "Set Bug Solved" (highlighted with a red oval), "Set Invalid", "Assign & Change Status", and "View Details".

6.1.6 Provide Comments (User Stories #8)

As a user, I want to be able to provide and see comments of a bug so that I can participate in the discussion of a certain bug.

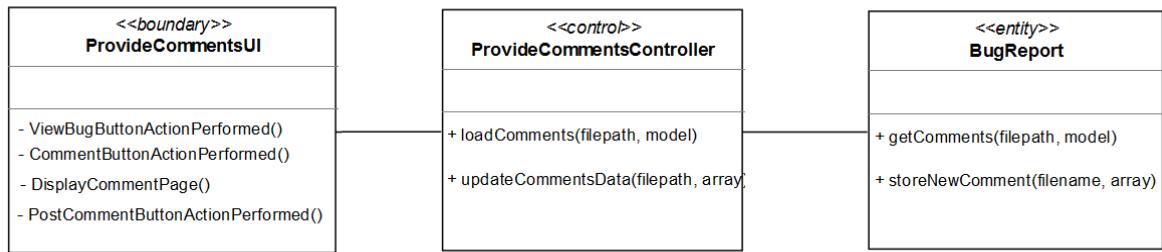
Use Case Diagram



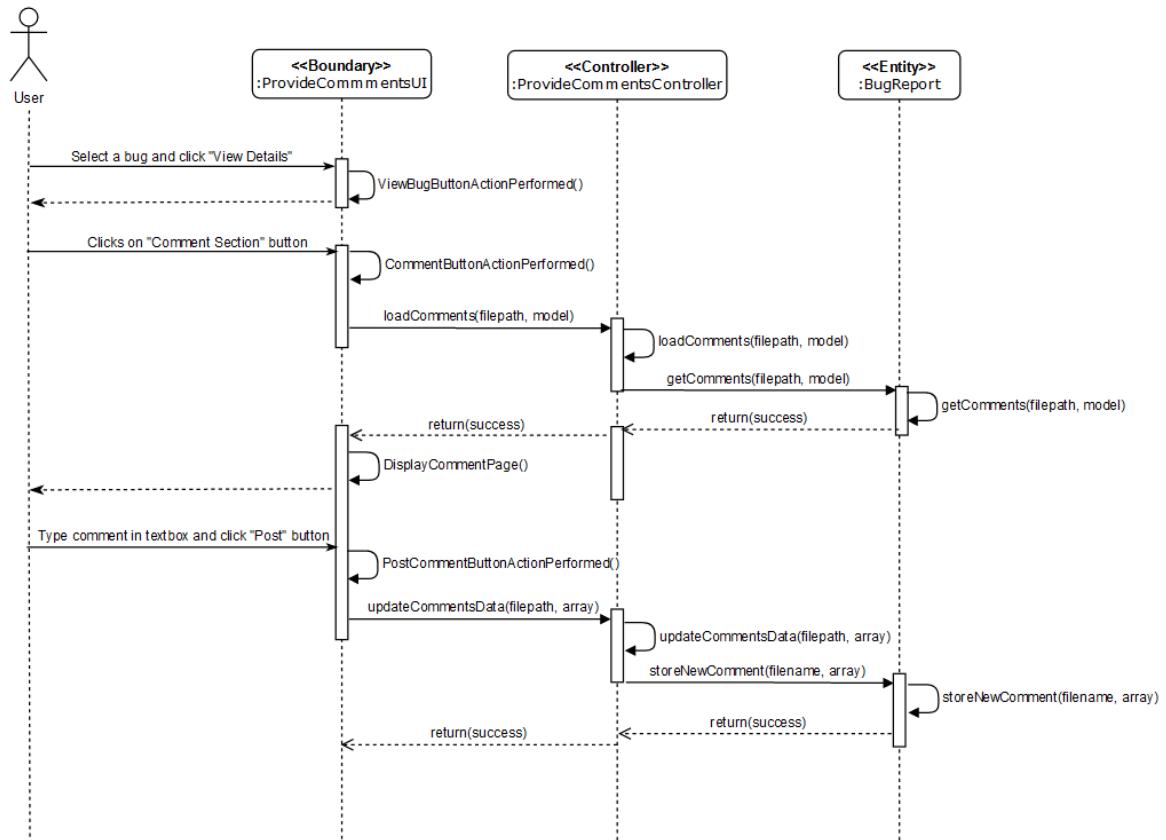
Use Case Description

Use Case Name: Provide Comments	ID: 8
Stakeholders and goals: User wants to be able to see and provide comments	
Description: A user wants to be able to provide comments so that he can participate in the discussion of a certain bug.	
Actors: User	
Trigger: The user selects the “Comment Section” button inside the Bug Specifications page. User can see all the comments for this bug if available and can post comments after clicking the “Post” button	
Preconditions: <ol style="list-style-type: none">1. User must be logged in to the system2. There must be bug reported in the system3. User must be inside the “Bug Specifications” page	
Normal flow: <ol style="list-style-type: none">1. System displays all the details of a bug2. User clicks the “Comment Section” button3. System displays the comment page with all the available comments4. User can type his comments in the textbox at the bottom and click the “Post” button to post the comment5. System will save and display the submitted comment in the comment page6. End	
Sub flows: None	
Alternatives/Exceptional flows: <ol style="list-style-type: none">4a. Back button: System will exit the comment page	

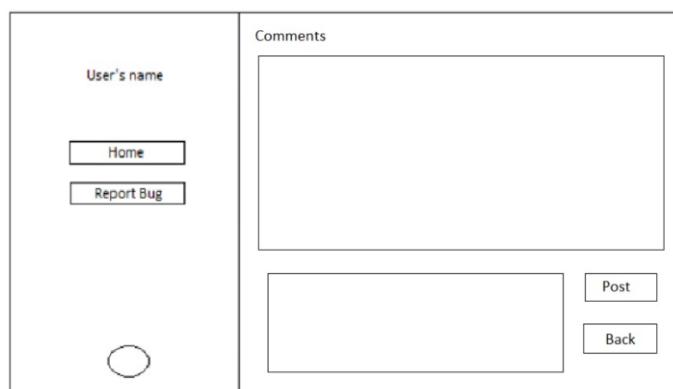
BCE Class Diagram



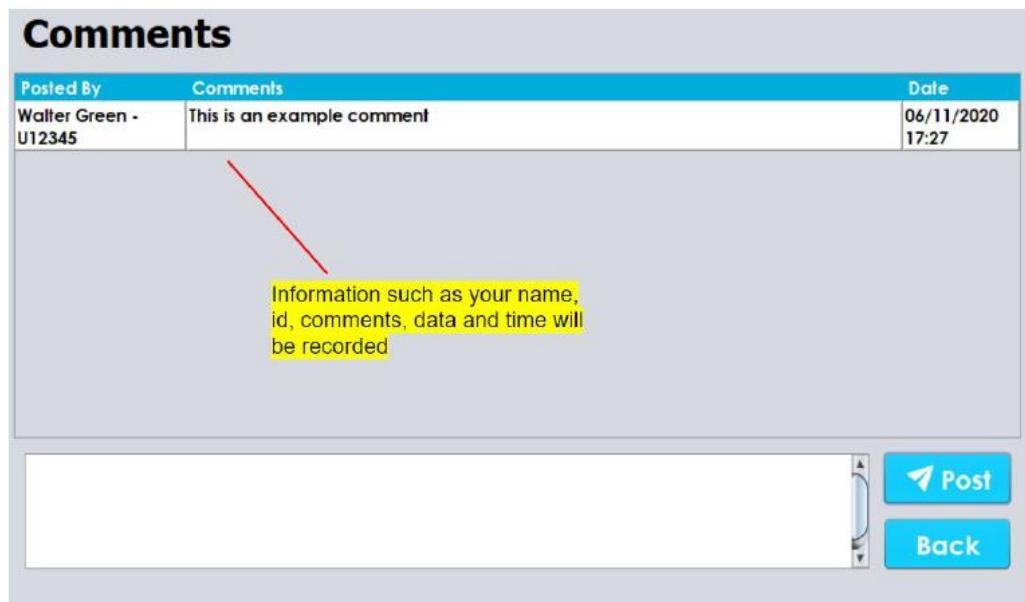
Sequence Diagram



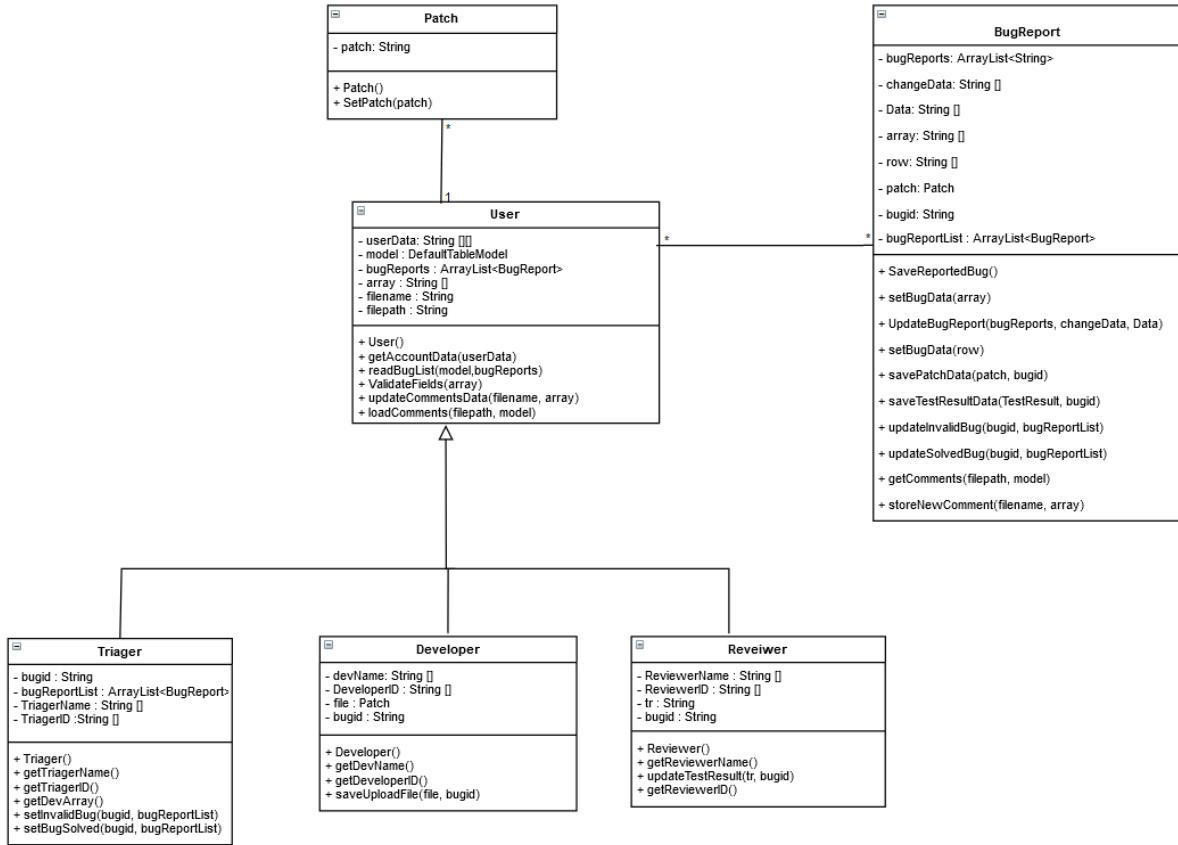
Wireframe



GUI



6.2 Class Diagram



6.3 Test Cases

6.3.1 Functional Test Case

View Assigned Bug

1	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	View Assigned Task function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	21/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	View all assigned bugs being assigned to developer	(1) Click on View Assigned Bugs button (2) Select a relevant bug (3) Click on View Details button	Shows all the details about a bug that is being assigned	Bug specifications page is displayed to the developer	Pass	NA	

Submit Fixed Bug

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Submit Fixed Bug function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	21/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successful submit a fixed bug - Developer	(1) Select a relevant bug (2) Click on Submit Patch button (3) Attach relevant file (4) Click on Submit button	Home page would be displayed with a popout message telling developer that the submission is successful	Home page is displayed to the developer	Pass	NA	

Input Test Result

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Input test result function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	21/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successful input test result - Reviewer	(1) Select a relevant bug (2) Click on Test Patch button (3) Input relevant testing results (4) Click on Submit Test button	Input Patch Test Results page would be displayed and reviewer can proceed to submit	Home page is displayed to the reviewer	Pass	NA	

Set Bug as Invalid

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Set bug as invalid function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	16/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successful change current status of bug report to invalid - Set by triager	(1) Select a relevant bug (2) Click on "Set Invalid" button	The current status of the bug report will be changed to invalid	Current status of the bug report is set to "Invalid"	Pass	NA	

Set Bug as Solved

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Set bug report as solved function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	16/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successful set the current status of the bug report to Solved - set by triager	(1) Select a relevant bug (2) Click on "Solved" button	The current status of the selected bug report will be set to "Resolved"	The current status of the selected bug report will be set to "Resolved"	Pass	NA	

Provide Comments

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Provide Comments function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	16/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successful providing comments - Can be done by all users	(1) Select a relevant bug (2) Click on View Details button (3) Click on Comment Section button (4) Input relevant comment (5) Click on Post Button	Comment written by user would be displayed on Comments page	Comment written by user is displayed on Comments page and can also view all the comments made by other users	Pass	NA	

6.4 Evidence of Use of Methodologies

6.4.1 Meeting Minute

Project Name: Bug tracking system

Meeting Objection: Prepare and begin sprint 3

Date: 15th October 2020

Attendee

Ng Ming Yao

Lim Li Shan

Neo Zhi Kai

Liu Zihan

Mallah Rahul Premchand

Chong Jun Wei

Ivan Lee You Qing

Agenda:

- Define what can be delivered in sprint 3
- Decide on the goal for Sprint 3
- Revise on Product Backlog
- Work on Sprint 3 Backlog

S/No	Item	Action Item	Due Date
1	Distribution of tasks	Task distributed by MingYao	15 th October 2020
2	Add/Remove user stories to backlog	To be done by team	15 th October 2020
3	Create sprint 3 backlog	Done by Zhi Kai	15 th October 2020
4	Complete all tasks in Sprint 3	Done by team	21 st October 2020

Date and time of next meeting:

Sprint 4: 22nd October 2020

Meeting minutes 3 prepared by Zhi Kai

6.4.2 Sprint Planning Meeting

Taiga - Product Backlog (Edited Items)

BACKLOG SIM2020Q4 - TEAM7268

32% 136 defined points 43 closed points 22 points / sprint			
SHOW FILTERS	SHOW TAGS	VELOCITY FORECASTING	
Votes	User Stories	Status	Points
<input type="checkbox"/>	▲ 0 #18 As a developer, I want to be able to check for my assigned task's status and problem reported so that I can begin fixing it.	New	12
<input type="checkbox"/>	▲ 0 #19 As a developer, I want to be able to submit my assigned task so that it can be tested by the reviewers.	New	9
<input type="checkbox"/>	▲ 0 #22 As a reviewer, I want to be able to input my test result so that triager could mark the bug as solved.	New	7
<input type="checkbox"/>	▲ 0 #13 As a triager, I want to be able to set a bug as invalid so that invalid bugs can be ignored.	New	6
<input type="checkbox"/>	▲ 0 #14 As a triager, I want to be able to set the status of a bug as solved so that it can show that the bug has been resolved.	New	7
<input type="checkbox"/>	▲ 0 #8 As a user, I want to be able to provide comments so that I can participate in the discussion of a certain bug.	New	8
<input type="checkbox"/>	▲ 0 #3 As a user, I want to be able to search a bug through keywords so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New	6
<input type="checkbox"/>	▲ 0 #6 As a user, I want to be able to search a bug through titles so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New	6
<input type="checkbox"/>	▲ 0 #4 As a user, I want to be able to search a bug through assignee so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New	6
<input type="checkbox"/>	▲ 0 #5 As a user, I want to be able to search a bug through reporter name so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New	6
<input type="checkbox"/>	▲ 0 #2 As a user, I want to be able to search a bug through bug ID so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New	6
<input type="checkbox"/>	▲ 0 #15 As a triager, I want to be able to generate a report that shows the total numbers of bugs reported in a selected timeline with all the information for each bug so that I can see the performance of reporters.	New	7
<input type="checkbox"/>	▲ 0 #16 As a triager, I want to be able to generate a report that shows the total numbers of bugs resolved in a selected timeline with all information for each bug so that I can see the performance of developers.	New	7

6.4.3 Taiga (End of Sprint)

SPRINT 3 SIM2020Q4 - TEA... 15 OCT 2020-21 OCT 2020

100% ✓ 49 total points 49 completed points 0 open tasks 54 closed tasks ↔ 0 locale closes					
USER STORY	NEW	IN PROGRESS	READY FOR TEST	CLOSED	NEEDS INFO
✗ #18 As a developer, I want to be able to check for my assigned task's status and problem reported so that I can begin fixing it. New 12points				#178 Draw Use Case Diagram for #18 As a developer #179 Create Use Case Description for #18 As a developer #182 B-C-E Design for #18 As a developer #181 Create Sequence Diagram for #18 As a developer #180 Create Class Diagram for #18 As a developer #183 Draw Wireframe for #18 As a developer #185 Code for #18 As a developer #184 Code GUI for #18 As a developer #186 Testing for #18 As a developer	
✗ #19 As a developer, I want to be able to submit my assigned task so that it can be tested by the reviewers. New 9points				#187 Draw Use Case Diagram for #19 As a developer #188 Create Use Case Description for #19 As a developer #191 B-C-E Design for #19 As a developer #190 Create Sequence Diagram for #19 As a developer #189 Create Class Diagram for #19 As a developer #192 Draw Wireframe for #19 As a developer #194 Code for #19 As a developer	

<p>✗ #22 As a reviewer, I want to be able to input my test result so that triager could mark the bug as solved.</p> <p>New 7points</p>	<p>+ </p>		<p>#193 Code GUI for #19 As a developer</p> <p>#195 Testing for for #19 As a developer</p>	#193 Code GUI for #19 As a developer #195 Testing for for #19 As a developer #41 Draw use case diagram for #22 As a reviewer #40 Create use case description for #22 As a reviewer #92 B-C-E Design for #22 As a reviewer #93 Create Sequence Diagram for #22 As a reviewer #91 Create Class Diagram for #22 As a reviewer #196 Draw Wireframe for #22 As a reviewer #197 Code GUI for #22 As a reviewer #198 Code for #22 As a reviewer #199 Testing for #22 As a reviewer
<p>✗ #13 As a triager, I want to be able to set a bug as invalid so that invalid bugs can be ignored.</p> <p>New 6points</p>	<p>+ </p>		#38 Draw use case diagram for #13 As a triager #39 Create use case description for #13 As a triager #95 B-C-E Design for #13 As a triager #96 Create Sequence Diagram for #13 As a triager #94 Create Class Diagram for #13 As a triager #200 Draw Wireframe for #13 As a triager #202 Code for #13 As a triager #201 Code GUI for #13 As a triager #203 Testing for #13 As a triager	#36 Draw use case diagram for #14 As a triager #37 Create use case description for #14 As a triager #98 B-C-E Design for #14 As a triager #99 Create Sequence Diagram for #14 As a triager #97 Create Class Diagram for #14 As a triager #204 Draw Wireframe for #14 As a triager #206 Code functionalities for #14 As a triager #205 Code GUI for #14 As a triager #207 Testing for #14 As a triager
<p>✗ #14 As a triager, I want to be able to set the status of a bug as solved so that it can show that the bug has been resolved.</p> <p>New 7points</p>	<p>+ </p>		#136 Draw Use Case Diagram for #8 As a user #137 Create Use Case Description for #8 As a user #139 B-C-E Design for #8 As a user #140 Create Sequence Diagram for #8 As a user	#138 Create Class Diagram for #8 As a user #208 Draw Wireframe for #8 As a user #209 Code GUI for #8 As a user #210 Code functionalities for #8 As a user #211 Testing for #8 As a user

6.4.4 Sprint Review Meeting

Work accomplished during Sprint 3:

- Revise and enhance existing product backlog items
- Team decided on the goal for sprint 3
- Completed all tasks for items selected from product backlog to sprint 3

6.4.5 Sprint Retrospective Meeting

Work to complete next reporting period:

- Preparation for Sprint 4
- Reprioritize features according to requirements

What went well in the Sprint:

- Goals and instructions for each individual are clear, team members are aware of the direction we are heading towards
- Team members are able to agree and accept on a similar goal thus making it easier for the project to be carried out

What can be improved:

- Programmers and tester needs communicate closely on how the developed program works

(Discussions)Suggestions/Issues:

- Programmers and tester should discuss each time a functionality has been implemented

6.4.6 Member Contribution

Name:	Task Completed
Ng Ming Yao	<ul style="list-style-type: none"> • Distribution of roles • Update class diagrams for sprint 3 user stories • Update B-C-E diagrams for sprint 3 user stories • Update sequence diagrams for sprint 3 user stories • Code GUI for sprint 3 user stories • Code functionalities for sprint 3 user stories
Lim Li Shan	<ul style="list-style-type: none"> • Create week 3 report • Create use case diagrams for sprint 3 user stories • Create use case descriptions for sprint 3 user stories • Update class diagrams for sprint 3 user stories • Create B-C-E diagrams for sprint 3 user stories • Create sequence diagrams for sprint 3 user stories
Neo Zhi Kai	<ul style="list-style-type: none"> • Create meeting minutes 3 • Create week 3 report • Create use case diagrams for sprint 3 user stories • Create use case descriptions for sprint 3 user stories • Update class diagrams for sprint 3 user stories • Create B-C-E diagrams for sprint 3 user stories • Create sequence diagrams for sprint 3 user stories • Analyzed and evaluated product backlog • Prepare sprint 3 backlog
Liu Zihan	<ul style="list-style-type: none"> • Create third batch of bug reports
Mallah Rahul Premchand	<ul style="list-style-type: none"> • Update class diagrams for sprint 3 user stories • Update B-C-E diagrams for sprint 3 user stories • Update sequence diagrams for sprint 3 user stories • Code GUI for sprint 3 user stories • Code functionalities for sprint 3 user stories
Chong Jun Wei	<ul style="list-style-type: none"> • Create and complete testing for sprint 3 user stories
Ivan Lee You Qing	<ul style="list-style-type: none"> • Create third batch of bug reports

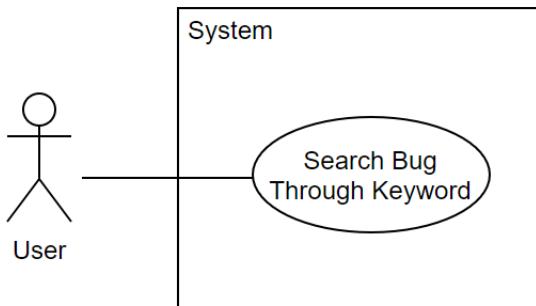
7 Sprint 4

7.1 User Stories and Diagrams

7.1.1 Search Bug Through Keyword (User Story #3)

As a user, I want to be able to search a bug through keywords so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.

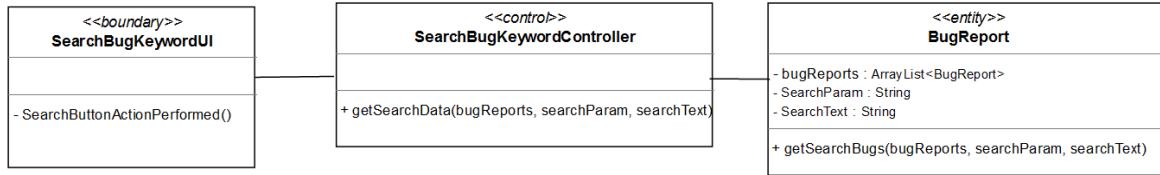
Use Case Diagram



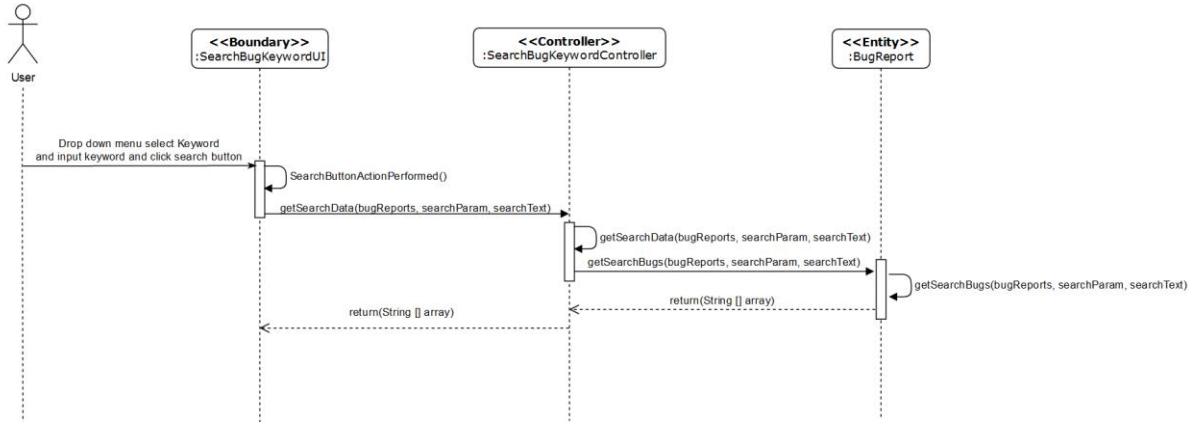
Use Case Description

Use Case Name: Search Bug Through Keyword	ID: 3
Stakeholders and goals: User wants to be able to search for a bug through keyword	
Description: A user is able to retrieve information from the search	
Actors: User	
Trigger: A user selects "Keyword" from the search drop down list and type the keyword he wants to search	
Preconditions: User must be logged in to the system	
Normal flow: <ol style="list-style-type: none">1. A user selects "Keyword" from the search drop down list2. User type the keyword he wants to search3. The user clicks on the search button4. System will verify the keyword and display ONLY bug reports that contains the keyword in the Dashboard (Bug ID, Bug Title, Keyword, Date, Description, Summary, Reporter, Assignee, Developer, Current Status, Priority, Severity, Patch, Reviewer, Test Results)5. End	
Sub flows: None	
Alternatives/Exceptional flows: 4: Empty display: No keyword matches the system	

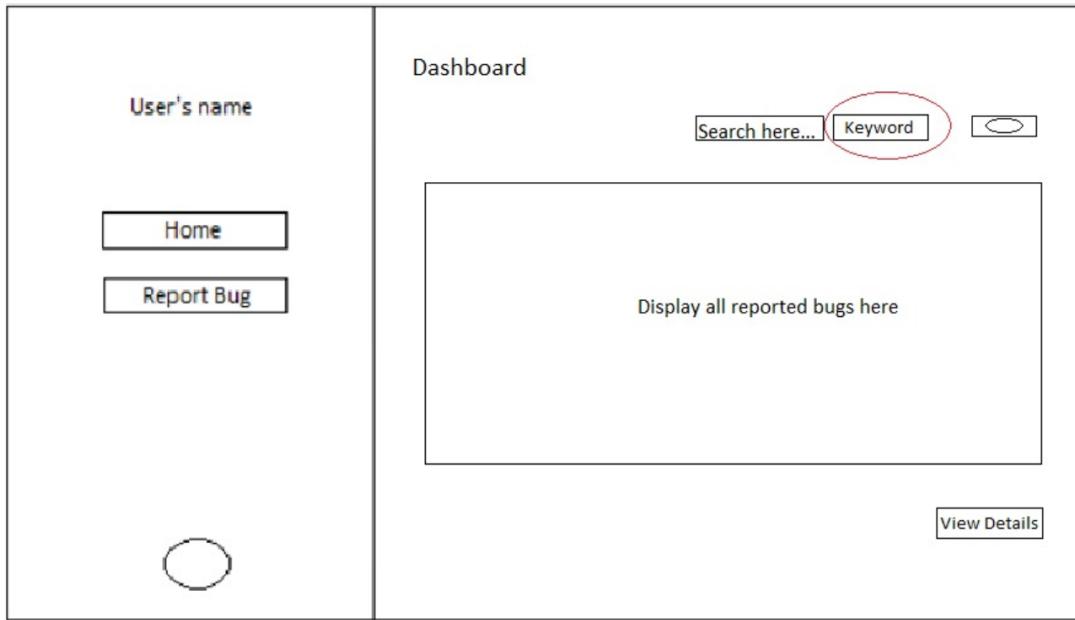
BCE Class Diagram



Sequence Diagram



Wireframe



GUI

Dashboard

(1): Type in a keyword (2): Select keyword as the search method (3): Click search

Bug ID	Bug Title	Keyword	Date	Description
10000	Application crash on cli...	Save button	11/10/2020	System crash when clicking SAVE
10001	Unable to delete user	Delete button	11/10/2020	Admin cannot delete user
10002	Username login caps sen...	Login	11/10/2020	User must type according to caps sen
10003	New accounts cannot ch...	Password	11/10/2020	User cannot change password
10004	System Crash when click...	Logout	11/10/2020	System crash when logging out
10005	Logout dialog box remains	Logout	11/10/2020	Shows logout but dialog box stays
10006	Unable to edit comment	Edit	11/10/2020	User cannot edit comment
10007	Comment edited but still...	Edit	11/10/2020	Edited comment is not shown
10008	No results found when se...	Search	11/10/2020	User cannot search
10009	Search results open in a ...	Search	11/10/2020	User search results in a new window
10010	Non-ASCII characters ge...	Search	12/10/2020	Error cause of non-ascii characters
10011	Advanced search functi...	Search	12/10/2020	Advanced search function not workin

[View Details](#)

Dashboard

Bug ID	Bug Title	Keyword	Date	Description
10008	No results found when se...	Search	11/10/2020	User cannot search
10009	Search results open in a ...	Search	11/10/2020	User search results in a new window
10010	Non-ASCII characters ge...	Search	12/10/2020	Error cause of non-ascii characters
10011	Advanced search functi...	Search	12/10/2020	Advanced search function not workin
10012	Search button shows eve...	Search	12/10/2020	Searching of specified bug reports yet
10013	Searching for reports ca...	Search	12/10/2020	User must type according to caps sen
10014	Search suggestions are ...	Search	12/10/2020	Search suggestions are missing
10015	Search history not saved	Search	12/10/2020	Search history cant be saved
10029	Error in showing the corre...	Search bug	13/10/2020	Error in the search engine.
10030	Error in displaying the co...	Search bug	14/10/2020	Error in the search engine.
10031	Error in displaying the co...	Search bug	14/10/2020	Error in the search engine.
10032	Error in displaying the co...	Search bug	14/10/2020	Error in the search engine.

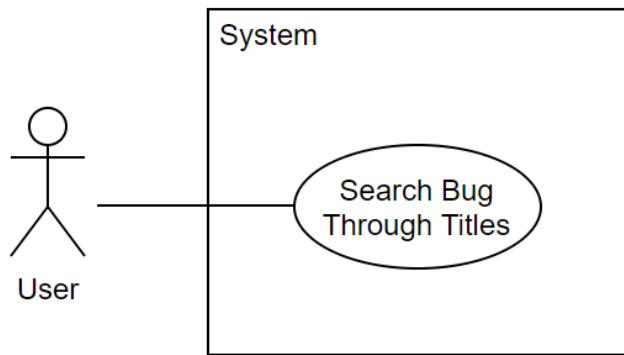
Bugs with the keyword will be displayed

[View Details](#)

7.1.2 Search Bug Through Titles (User Story #6)

As a user, I want to be able to search a bug through titles so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.

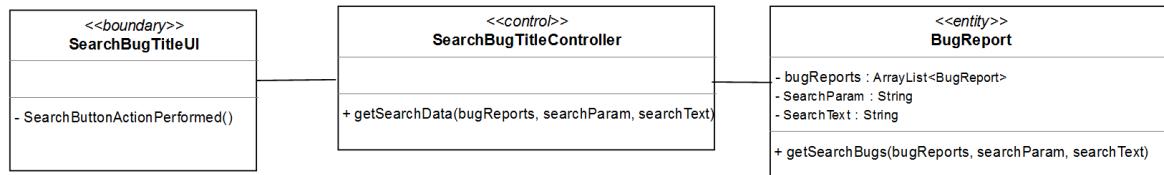
Use Case Diagram



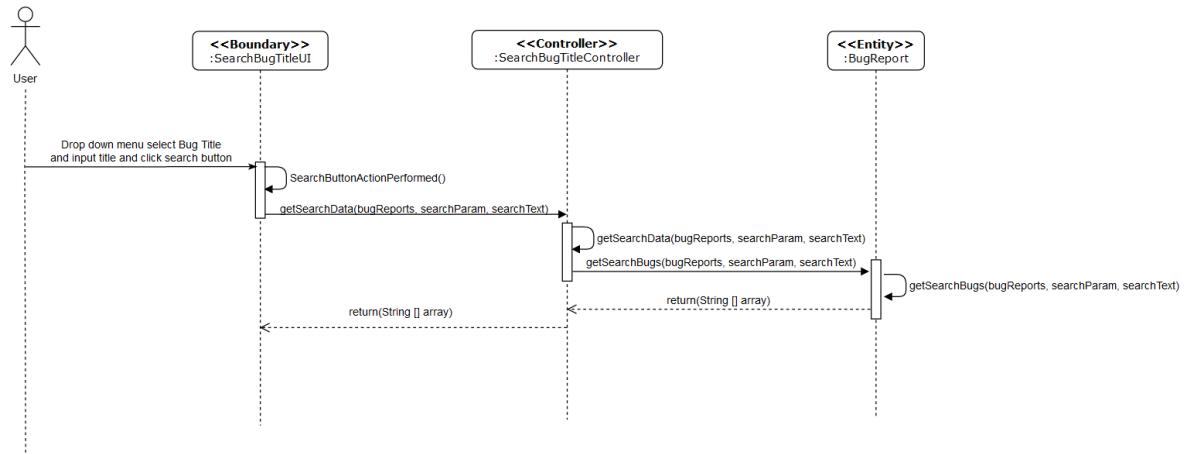
Use Case Description

Use Case Name: Search Bug Through Titles	ID: 6
Stakeholders and goals: User wants to be able to search for a bug through titles	
Description: A user is able to retrieve information from the search	
Actors: User	
Trigger: A user selects "Bug Title" from the search drop down list and type the title he wants to search	
Preconditions: User must be logged in to the system	
Normal flow: <ol style="list-style-type: none">1. A user selects "Bug Title" from the search drop down list2. User type the bug title he wants to search3. The user clicks on the search button4. System will verify the bug titles and display ONLY all of a bug information such as the bug's ID, bug titles, reporter name, issue description, summary, reported date, assigned developer, current status, Severity, Priority and Comments that matches the bug titles, in the home page5. End	
Sub flows: None	
Alternatives/Exceptional flows: 4: Empty display: No bug titles matches the system	

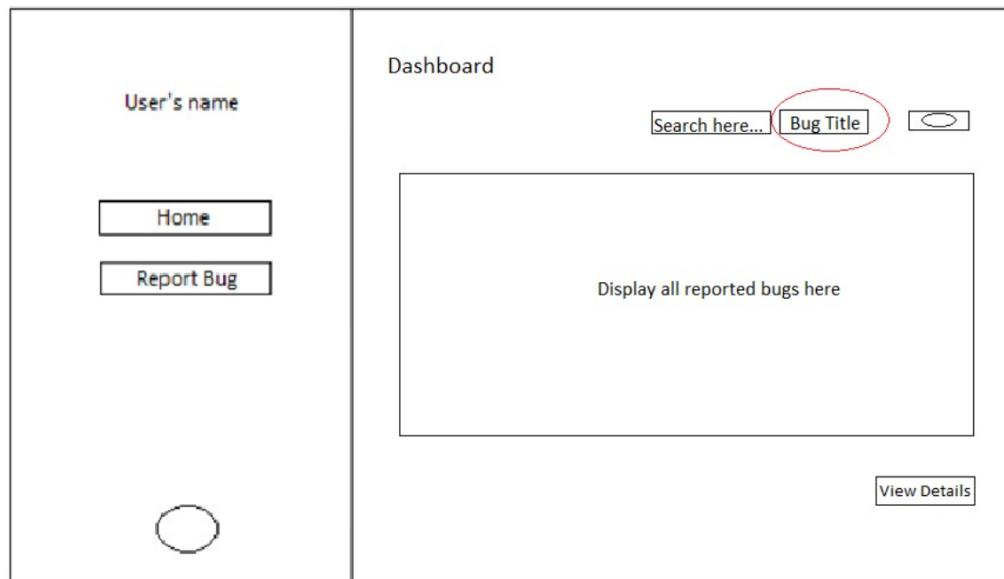
BCE Class Diagram



Sequence Diagram



Wireframe



GUI

The screenshot shows a Windows application window titled "Dashboard". At the top, there is a search bar with a placeholder "Error" and a dropdown menu set to "Bug Title". To the right of the dropdown is a magnifying glass icon representing the search button. A yellow callout box labeled "(1): Type in a bug title" points to the search input field. Another yellow callout box labeled "(2): Select bug title as the search method" points to the dropdown menu. A third yellow callout box labeled "(3): Click search" points to the magnifying glass icon. Below the search bar is a table listing 11 bugs. The columns are "Bug ID", "Bug Title", "Keyword", "Date", and "Description". The bugs listed are:

Bug ID	Bug Title	Keyword	Date	Description
10000	Application crash on cli...	Save button	11/10/2020	System crash when clicking SAVE
10001	Unable to delete user	Delete button	11/10/2020	Admin cannot delete user
10002	Username login caps sen...	Login	11/10/2020	User must type according to caps sen
10003	New accounts cannot ch...	Password	11/10/2020	User cannot change password
10004	System Crash when click...	Logout	11/10/2020	System crash when logging out
10005	Logout dialog box remains	Logout	11/10/2020	Shows logout but dialog box stays
10006	Unable to edit comment	Edit	11/10/2020	User cannot edit comment
10007	Comment edited but still...	Edit	11/10/2020	Edited comment is not shown
10008	No results found when se...	Search	11/10/2020	User cannot search
10009	Search results open in a ...	Search	11/10/2020	User search results in a new window
10010	Non-ASCII characters ge...	Search	12/10/2020	Error cause of non-ascii characters
10011	Advanced search functi...	Search	12/10/2020	Advanced search function not workin

A blue button at the bottom right of the table area is labeled "View Details".

This screenshot shows the same "Dashboard" application window. The search bar now contains the text "Error". The dropdown menu is still set to "Bug Title". The magnifying glass icon is visible to the right of the dropdown. A yellow callout box labeled "Bug with the bug title will be displayed" points to the table area. The table lists 13 bugs, all of which have "Error" in their titles. The first few rows of the table are:

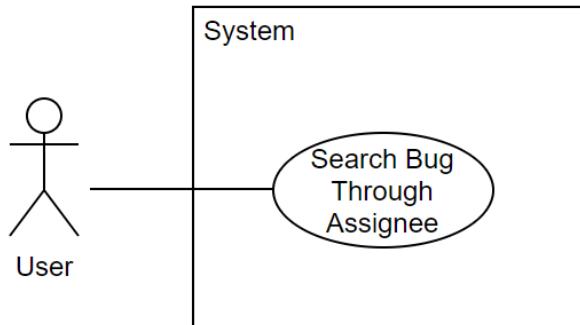
Bug ID	Bug Title	Keyword	Date	Description
10022	Error in saving the bug titl...	Report bug	13/10/2020	Error in saving the bug title information
10023	Error in generating the re...	Report bug	13/10/2020	Error in generating the correct date
10024	Error in saving the bug d...	Report bug	13/10/2020	Error in saving the bug description
10025	Error in generating a bug...	Report bug	13/10/2020	Error in generating the bug ID
10026	Error in generating the re...	Report bug	13/10/2020	Error in generating the reporter's in
10027	Error in generating the re...	Report bug	13/10/2020	Error in generating the reporter's inform
10028	Error in saving the bug inf...	Report bug	13/10/2020	Bug reported not saved in the database
10029	Error in showing the corre...	Search bug	13/10/2020	Error in the search engine.
10030	Error in displaying the co...	Search bug	14/10/2020	Error in the search engine.
10031	Error in displaying the co...	Search bug	14/10/2020	Error in the search engine.
10032	Error in displaying the co...	Search bug	14/10/2020	Error in the search engine.
10033	Error in assigning the bug...	Assign bug	14/10/2020	The system couldn't assign the bu

A blue button at the bottom right of the table area is labeled "View Details".

7.1.3 Search Bug Through Assignee (User Stories #4)

As a user, I want to be able to search a bug through assignee so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.

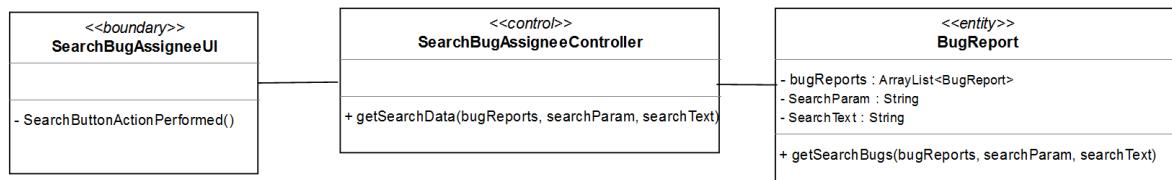
Use Case Diagram



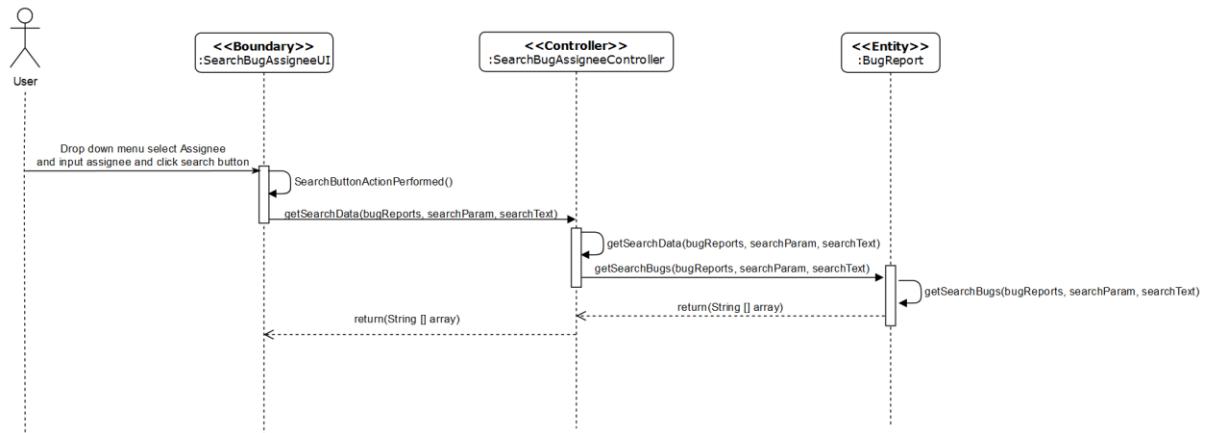
Use Case Description

Use Case Name: Search Bug Through Assignee	ID: 4
Stakeholders and goals: User want to be able to search for a bug through an assignee's name	
Description: A user is able to retrieve information from the search	
Actors: User	
Trigger: A user selects "Assignee" from the search drop down list and type the assignee name he wants to search	
Preconditions: User must be logged in to the system	
Normal flow: <ol style="list-style-type: none">1. A user selects "Assignee" from the search drop down list2. User type the assignee name he wants to search3. The user clicks on the search button4. System will verify the assignee's name and display ONLY all of a bug information such as the bug's ID, bug titles, reporter name, issue description, summary, reported date, assigned developer, current status, Severity, Priority and Comments that matches the assignee's name, in the home page5. End	
Sub flows: None	
Alternatives/Exceptional flows: 4: Empty display: No bug being assigned to this assignee's name at that point of time or because there is no such assignee's name in the system	

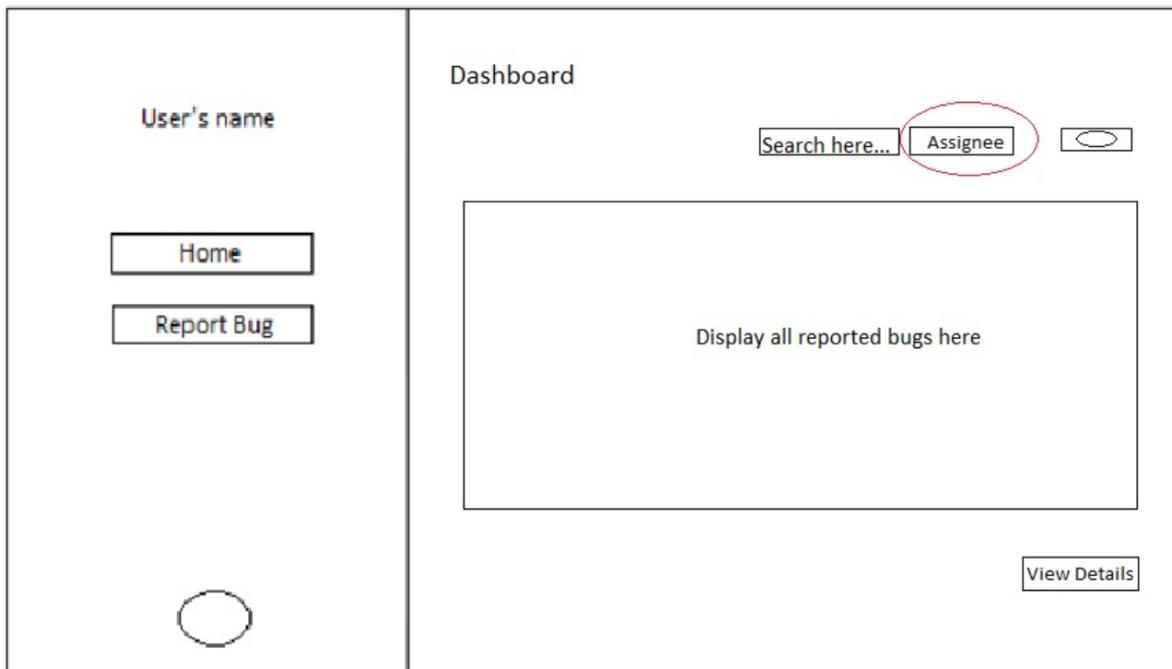
BCE Class Diagram



Sequence Diagram



Wireframe



GUI

The screenshot shows a bug tracking system interface with a search function and a results table.

Search Function:

- (1): Type in a Assignee name: A yellow callout points to the search input field containing "Chloe".
- (2): Select assignee as the search method: A yellow callout points to the dropdown menu labeled "Assignee".
- (3): Click search: A yellow callout points to the magnifying glass icon button.

Results Table:

Reporter	Assignee	Developer	Current Status
vn when they creating a...	Lance Hanson - U1...	Chloe Dillard - T123...	Lacey Best - D12351
8	Caleb Rodriguez - ...	Chloe Dillard - T123...	Venus Maldonado ...
search box after typing	Dominic Newman - ...	Chloe Dillard - T123...	David White - D12345
found' is not shown at t...	Dominique Crane - ...	Chloe Dillard - T123...	Phillip Cannon - D1...
ch the bugs that has hig...	Dominique Crane - ...	Chloe Dillard - T123...	Carl Higgins - D12352
ug reports" but search re...	Beck Oneal - U12358	Chloe Dillard - T123...	Quinlan Walsh - D1...
to type in their email as ...	Walter Green - U12...	Chloe Dillard - T123...	Lacey Best - D12351
own "Jan, Feb, Mar etc&#...	Dominique Crane - ...	Chloe Dillard - T123...	Carl Higgins - D12352
use up/down button to n...	Daryl Keith - U12359	Chloe Dillard - T123...	David White - D12345
mouse over dropdown b...	Doris Pugh - U12349	Chloe Dillard - T123...	Cyrus Holt - D12346
ided to user	Dominic Newman - ...	Chloe Dillard - T123...	Cyrus Holt - D12346
vn when they creating a...	Lance Hanson - U1...	Chloe Dillard - T123...	Lacey Best - D12351

Buttons:

- View Details: A blue button located at the bottom right of the table.

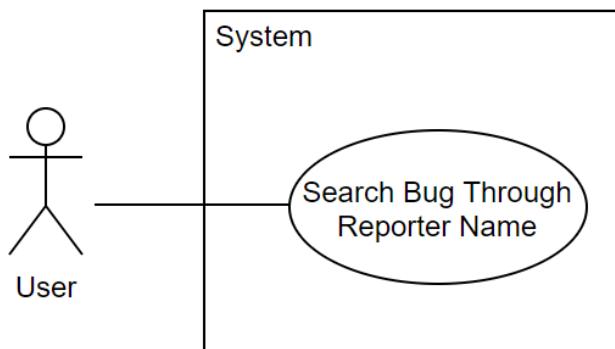
Callout:

Bugs with the selected assignee will be displayed

7.1.4 Search Bug Through Reporter Name (User Story #5)

As a user, I want to be able to search a bug through reporter name so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.

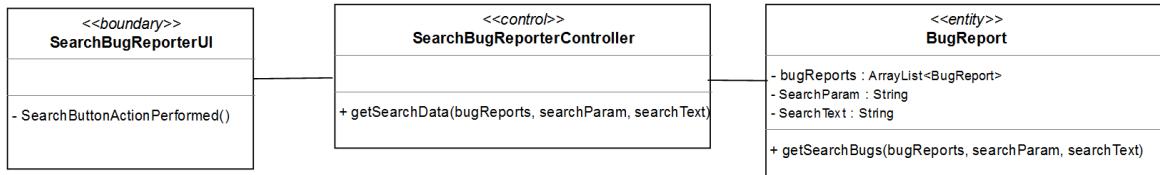
Use Case Diagram



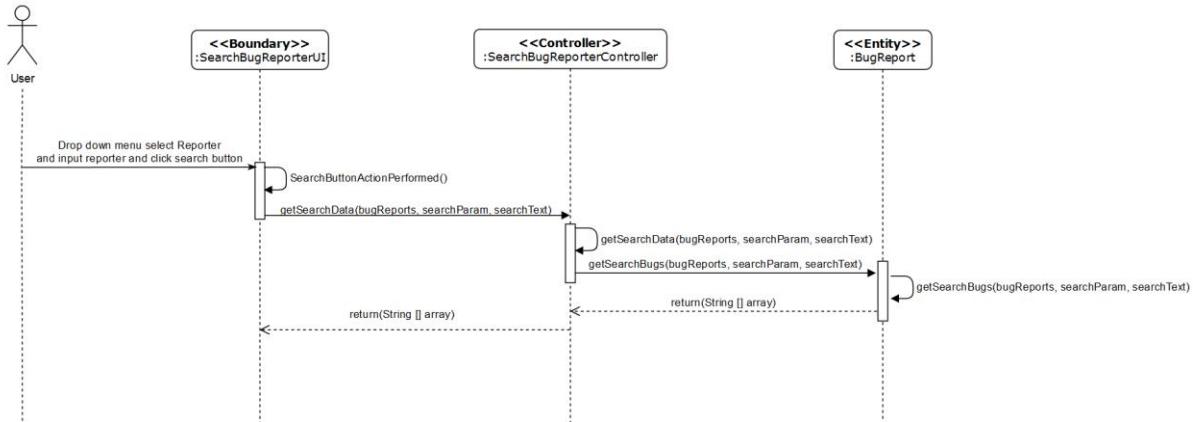
Use Case Description

Use Case Name: Search Bug Through Reporter Name	ID: 5
Stakeholders and goals: User want to be able to search for a bug through a reporter's name	
Description: A user is able to retrieve information from the search	
Actors: User	
Trigger: A user selects "Reporter" from the search drop down list and type the reporter name he wants to search	
Preconditions: User must be logged in to the system	
Normal flow: <ol style="list-style-type: none">1. A user selects "Reporter" from the search drop down list2. User type the reporter name he wants to search3. The user clicks on the search button4. System will verify the reporter's name and display ONLY all of a bug information such as the bug's ID, bug titles, reporter name, issue description, summary, reported date, assigned developer, current status, Severity, Priority and Comments that matches the reporter's name, in the home page5. End	
Sub flows: None	
Alternatives/Exceptional flows: <p>4: Empty display: No bug reported by this reporter's name at that point of time or because there is no such reporter's name in the system</p>	

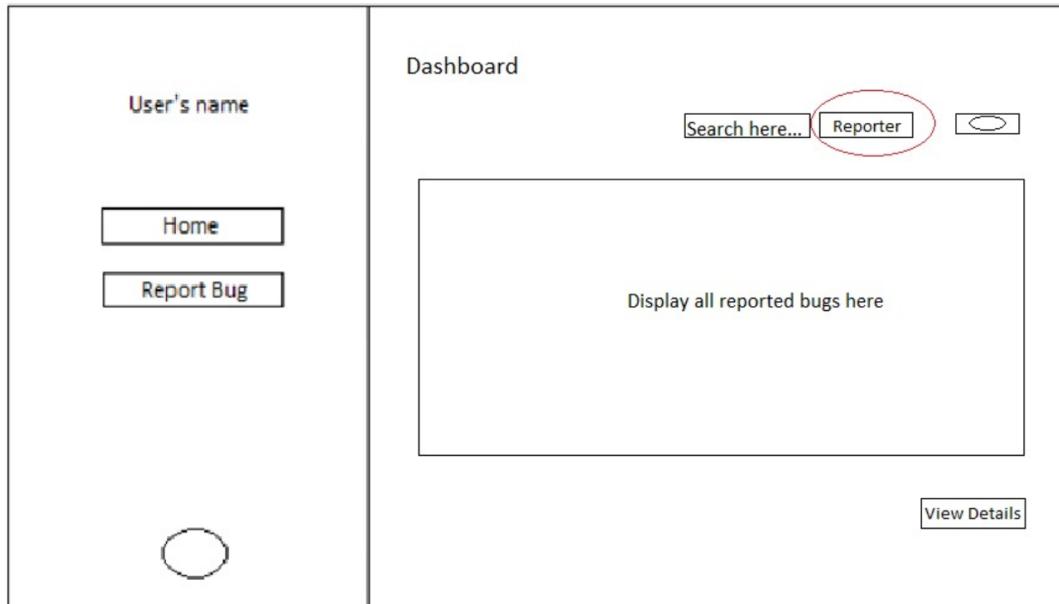
BCE Class Diagram



Sequence Diagram



Wireframe



GUI

(1): Type in a Reporter name
(2): Select reporter as the search method
(3): Click search

Bug ID	Bug Title	Keyword	Date	Description
10021	Identification of the user ...	User type	13/10/2020	The system is not able to recognize th...
10022	Error in saving the bug titl...	Report bug	13/10/2020	Error in saving the bug title information
10023	Error in generating the re...	Report bug	13/10/2020	Error in generating the correct date
10024	Error in saving the bug d...	Report bug	13/10/2020	Error in saving the bug description
10025	Error in generating a bug...	Report bug	13/10/2020	Error in generating the bug ID
10026	Error in generating the re...	Report bug	13/10/2020	Error in generating the reporterâ€™s infor...
10027	Error in generating the re...	Report bug	13/10/2020	Error in generating the reporter's inform...
10028	Error in saving the bug inf...	Report bug	13/10/2020	Bug reported not saved in the database
10029	Error in showing the corre...	Search bug	13/10/2020	Error in the search engine.
10030	Error in displaying the co...	Search bug	14/10/2020	Error in the search engine.
10031	Error in displaying the co...	Search bug	14/10/2020	Error in the search engine.
10032	Error in displaying the co...	Search bug	14/10/2020	Error in the search engine.

View Details

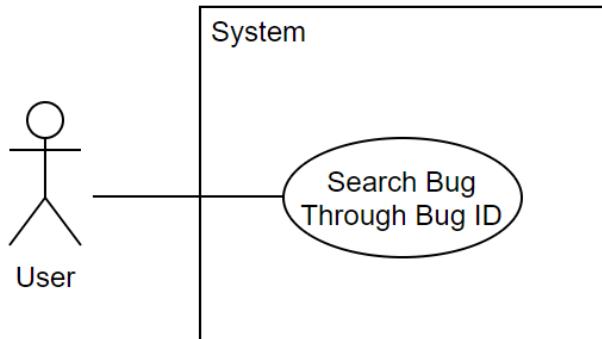
Bugs with the stated reporter name will be displayed

View Details

7.1.5 Search Bug Through Bug ID (User Story #2)

As a user, I want to be able to search a bug through bug ID so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.

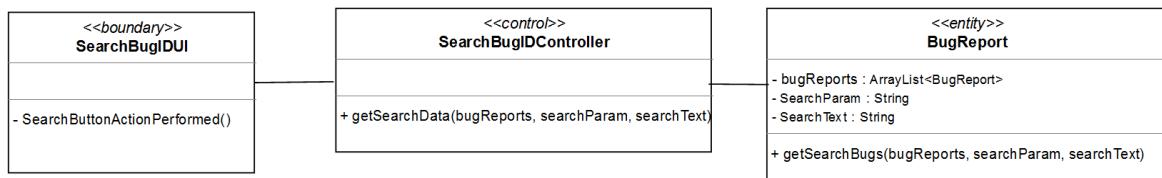
Use Case Diagram



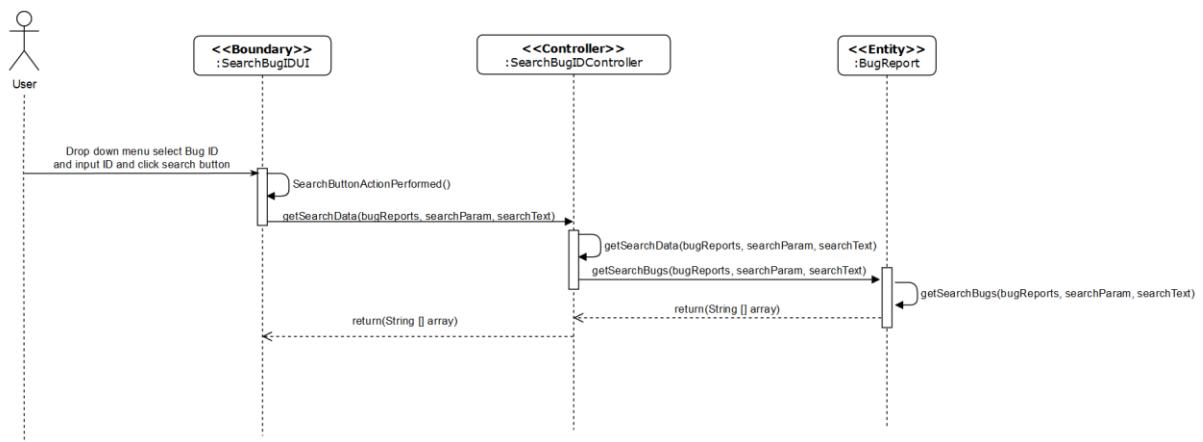
Use Case Description

Use Case Name: Search Bug Through Bug ID	ID: 2
Stakeholders and goals: User wants to be able to search for a bug through a Bug ID	
Description: A user is able to retrieve information from the search	
Actors: All user	
Trigger: A user selects "Bug ID" from the search drop down list and type the reporter name he wants to search	
Preconditions: User must be logged in to the system	
Normal flow: <ol style="list-style-type: none">1. A user selects "Bug ID" from the search drop down list2. User type the Bug ID he wants to search3. The user clicks on the search button4. System will verify the Bug ID and display ONLY all of a bug information such as the bug's ID, bug titles, reporter name, issue description, summary, reported date, assigned developer, current status, Severity, Priority and Comments that matches the Bug ID, in the home page5. End	
Sub flows: None	
Alternatives/Exceptional flows: 4: Empty display: No bug assigned with this Bug ID at that point of time	

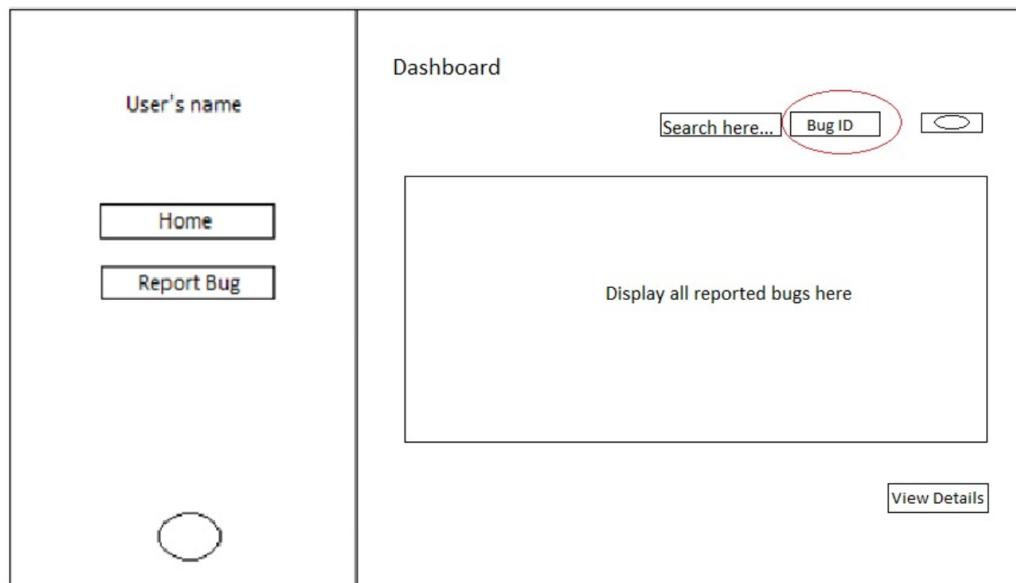
BCE Class Diagram



Sequence Diagram



Wireframe



GUI

Dashboard

(1): Type in a bug ID
10022

(2): Select bug ID as the search method
Bug ID

(3): Click search

Bug ID	Bug Title	Keyword	Date	Description
10018	Bug report does not refle...	Bug reports	12/10/2020	Bug reports not shown instantly
10019	Unable to edit bug report	Bug reports	12/10/2020	User cannot edit bug report
10020	Password not verified	Password	13/10/2020	The system does not verify the passwo
10021	Identification of the user ...	User type	13/10/2020	The system is not able to recognize th
10022	Error in saving the bug titl...	Report bug	13/10/2020	Error in saving the bug title information
10023	Error in generating the re...	Report bug	13/10/2020	Error in generating the correct date
10024	Error in saving the bug d...	Report bug	13/10/2020	Error in saving the bug description
10025	Error in generating a bug...	Report bug	13/10/2020	Error in generating the bug ID
10026	Error in generating the re...	Report bug	13/10/2020	Error in generating the reporterâ€™s in
10027	Error in generating the re...	Report bug	13/10/2020	Error in generating the reporter's inform
10028	Error in saving the bug inf...	Report bug	13/10/2020	Bug reported not saved in the database
10029	Error in showing the corre...	Search bug	13/10/2020	Error in the search engine.

[View Details](#)

Dashboard

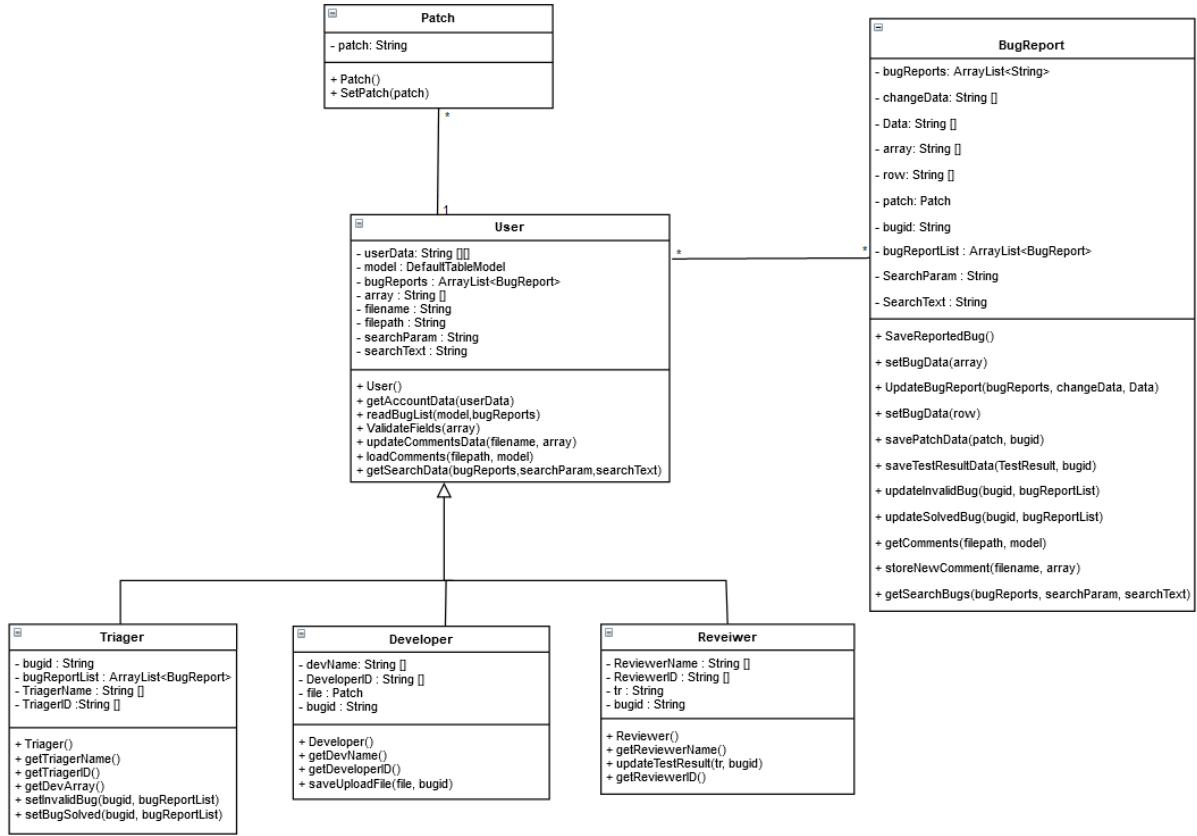
10022

Bug ID

Bug ID	Bug Title	Keyword	Date	Description
10022	Error in saving the bug titl...	Report bug	13/10/2020	Error in saving the bug title information

Selected bug information will be displayed

7.2 Class Diagram



7.3 Test Cases

7.3.1 Functional Test Case

Search Bug Through Keyword

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Search Bug Through Keyword function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	28/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successfully search through Keyword	(1) Select a relevant bug (2) Select Keyword in the drop down menu (3) Type in bug's Keyword (4) Click on Search Icon	A pop up message would appear telling user that search is successful with the search result displayed	Dashboard page is displayed with the search result	Pass	NA	

Search Bug Through Titles

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Search Bug through Bug Title function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	28/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successfully search through Bug Title	(1) Select a relevant bug (2) Select Bug Title in the drop down menu (3) Type in bug's Bug Title (4) Click on Search Icon	A pop up message would appear telling user that search is successful with the search result displayed	Dashboard page is displayed with the search result	Pass	NA	

Search Bug Through Assignee

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Search Bug Through Assignee function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	28/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successfully search through Assignee	(1) Select a relevant bug (2) Select Assignee in the drop down menu (3) Type in bug's Assignee (4) Click on Search Icon	A pop up message would appear telling user that search is successful with the search result displayed	Dashboard page is displayed with the search result	Pass	NA	

Search Bug Through Reporter Name

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Search Bug Through Reporter function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	28/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successfully search through Reporter	(1) Select a relevant bug (2) Select Reporter in the drop down menu (3) Type in bug's Reporter (4) Click on Search Icon	A pop up message would appear telling user that search is successful with the search result displayed	Dashboard page is displayed with the search result	Pass	NA	

Search Bug Through Bug ID

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case Description:	Search Bug Through Bug ID function						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	28/10/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successfully search through Bug ID	(1) Select a relevant bug (2) Select Bug ID in the drop down menu (3) Type in bug's Bug ID (4) Click on Search Icon	A pop up message would appear telling user that search is successful with the search result displayed	Dashboard page is displayed with the search result	Pass	NA	

7.4 Evidence of Use of Methodologies

7.4.1 Meeting Minute

Project Name: Bug tracking system

Meeting Objection: Prepare and begin sprint 4

Date: 22th October 2020

Attendee
Ng Ming Yao
Lim Li Shan
Neo Zhi Kai
Liu Zihan
Mallah Rahul Premchand
Chong Jun Wei
Ivan Lee You Qing

Agenda:

- Define what can be delivered in sprint 4
- Decide on the goal for Sprint 4
- Revise on Product Backlog
- Work on Sprint 4 Backlog

S/No	Item	Action Item	Due Date
1	Distribution of tasks	Task distributed by MingYao	22 nd October 2020
2	Add/Remove user stories to backlog	To be done by team	22 nd October 2020
3	Create sprint 4 backlog	Done by Zhi Kai	22 nd October 2020
4	Complete all tasks in Sprint 4	Done by team	28 th October 2020

Date and time of next meeting:

Sprint 5 (Final Sprint): 29th October 2020

Meeting minutes 4 prepared by Zhi Kai

7.4.2 Sprint Planning Meeting

Taiga - Product Backlog (Edited Items)

BACKLOG SIM2020Q4 - TEAM7268

	68%	136 defined points	92 closed points	31 points / sprint
SHOW FILTERS SHOW TAGS VELOCITY FORECASTING + ADD A NEW USER STORY				
Votes	User Stories	Status	Points	▼

	▲ 0 #3 As a user, I want to be able to search a bug through keywords so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6	
	▲ 0 #6 As a user, I want to be able to search a bug through titles so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6	
	▲ 0 #4 As a user, I want to be able to search a bug through assignee so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6	
	▲ 0 #5 As a user, I want to be able to search a bug through reporter name so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6	
	▲ 0 #2 As a user, I want to be able to search a bug through bug ID so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.	New ▾	6	
	▲ 0 #15 As a triager, I want to be able to generate a report that shows the total numbers of bugs reported in a selected timeline with all the information for each bug so that I can see the performance of reporters.	New ▾	7	
	▲ 0 #16 As a triager, I want to be able to generate a report that shows the total numbers of bugs resolved in a selected timeline with all information for each bug so that I can see the performance of developers.	New ▾	7	

7.4.3 Taiga (End of Sprint)

SPRINT 4 SIM2020Q4 - TEA... 22 OCT 2020-28 OCT 2020

	100% ▾	30 total points	30 completed points	0 open tasks	45 closed tasks			0 localine doses
USER STORY NEW IN PROGRESS READY FOR TEST CLOSED NEEDS INFO								
× #3 As a user, I want to be able to search a bug through keywords so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments. New 6points					+	#100 Draw Use Case Diagram for #3 As a user #213 Code GUI for #3 As a user #214 Code functionalities for #3 As a user #215 Testing for #3 As a user #212 Draw Wireframe for #3 As a user #101 Create Use Case Description for #3 As a user #102 Create Class Diagram for #3 As a user #104 Create Sequence Diagram for #3 As a user #103 B-C-E Design for #3 As a user		
× #6 As a user, I want to be able to search a bug through titles so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments. New 6points					+	#110 Draw Use Case Diagram for #6 As a user #217 Code GUI for #6 As a user #218 Code functionalities for #6 As a user #114 Create Sequence Diagram for #6 As a user #219 Testing for #6 As a user #216 Draw Wireframe for #6 As a user #111 Create Use Case Description for #6 As a user #113 B-C-E Design for #6 As a user #112 Create Class Diagram for #6 As a user		

<p><input checked="" type="checkbox"/> #4 As a user, I want to be able to search a bug through assignee so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.</p> <p>New 6points</p>	<p><input checked="" type="checkbox"/> #5 As a user, I want to be able to search a bug through reporter name so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.</p> <p>New 6points</p>	<p><input checked="" type="checkbox"/> #2 As a user, I want to be able to search a bug through bug ID so that I can track down a specific bug's ID, description, summary, reporter name, reported date, assigned developer, current status, Severity, Priority and Comments.</p> <p>New 6points</p>	<p><input checked="" type="checkbox"/> #105 Draw Use Case Diagram for #4 As a user</p> <p><input checked="" type="checkbox"/> #106 Create Use Case Description for #4 As a user</p> <p><input checked="" type="checkbox"/> #222 Code functionalities for #4 As a user</p> <p><input checked="" type="checkbox"/> #221 Code GUI for #4 As a user</p> <p><input checked="" type="checkbox"/> #223 Testing for #4 As a user</p> <p><input checked="" type="checkbox"/> #220 Draw Wireframe for #4 As a user</p> <p><input checked="" type="checkbox"/> #109 Create Sequence Diagram for #4 As a user</p> <p><input checked="" type="checkbox"/> #107 Create Class Diagram for #4 As a user</p> <p><input checked="" type="checkbox"/> #108 B-C-E Design for #4 As a user</p> <p><input checked="" type="checkbox"/> #120 Draw Use Case Diagram for #5 As a user</p> <p><input checked="" type="checkbox"/> #121 Create Use Case Description for #5 As a user</p> <p><input checked="" type="checkbox"/> #226 Code functionalities for #5 As a user</p> <p><input checked="" type="checkbox"/> #225 Code GUI for #5 As a user</p> <p><input checked="" type="checkbox"/> #227 Testing for #5 As a user</p> <p><input checked="" type="checkbox"/> #224 Draw Wireframe for #5 As a user</p> <p><input checked="" type="checkbox"/> #124 Create Sequence Diagram for #5 As a user</p> <p><input checked="" type="checkbox"/> #123 B-C-E Design for #5 As a user</p> <p><input checked="" type="checkbox"/> #122 Create Class Diagram for #5 As a user</p> <p><input checked="" type="checkbox"/> #125 Draw Use Case Diagram for #2 As a user</p> <p><input checked="" type="checkbox"/> #229 Code GUI for #2 As a user</p> <p><input checked="" type="checkbox"/> #230 Code functionalities for #2 As a user</p> <p><input checked="" type="checkbox"/> #231 Testing for #2 As a user</p> <p><input checked="" type="checkbox"/> #228 Draw Wireframe for #2 As a user</p> <p><input checked="" type="checkbox"/> #128 B-C-E Design for #2 As a user</p> <p><input checked="" type="checkbox"/> #129 Create Sequence Diagram for #2 As a user</p> <p><input checked="" type="checkbox"/> #127 Create Class Diagram for #2 As a user</p> <p><input checked="" type="checkbox"/> #126 Create Use Case Description for #2 As a user</p>
---	--	---	---

7.4.4 Sprint Review Meeting

Work accomplished during Sprint 4:

- Revise and enhance existing product backlog items
- Team decided on the goal for sprint 4
- Completed all tasks for items selected from product backlog to sprint 4

7.4.5 Sprint Retrospective Meeting

Work to complete next reporting period:

- Preparation for Sprint 5
- Reprioritize features according to requirements

What went well in the Sprint:

- Goals and instructions for each individual are clear, team members are aware of the direction we are heading towards
- Team members are able to agree and accept on a similar goal thus making it easier for the project to be carried out

What can be improved:

- Programmers and tester needs communicate closely on how the developed program works

(Discussions)Suggestions/Issues:

- Programmers and tester should discuss each time a functionality has been implemented

7.4.6 Member Contribution

Name:	Task Completed	Contribution
Ng Ming Yao	<ul style="list-style-type: none"> • Distribution of roles • Update class diagrams for sprint 4 user stories • Update B-C-E diagrams for sprint 4 user stories • Update sequence diagrams for sprint 4 user stories • Code GUI for sprint 4 user stories • Code functionalities for sprint 4 user stories 	
Lim Li Shan	<ul style="list-style-type: none"> • Create week 4 report • Create use case diagrams for sprint 4 user stories • Create use case descriptions for sprint 4 user stories • Update class diagrams for sprint 4 user stories • Create B-C-E diagrams for sprint 4 user stories • Create sequence diagrams for sprint 4 user stories 	
Neo Zhi Kai	<ul style="list-style-type: none"> • Create meeting minutes 4 • Create week 4 report • Create use case diagrams for sprint 4 user stories • Create use case descriptions for sprint 4 user stories • Update class diagrams for sprint 4 user stories • Create B-C-E diagrams for sprint 4 user stories • Create sequence diagrams for sprint 4 user stories • Analyzed and evaluated product backlog • Prepare sprint 4 backlog 	
Liu Zihan	<ul style="list-style-type: none"> • Create user manual for all sprints • Preparation for demo video • Assist in Testing 	
Mallah Rahul Premchand	<ul style="list-style-type: none"> • Update class diagrams for sprint 4 user stories • Update B-C-E diagrams for sprint 4 user stories • Update sequence diagrams for sprint 4 user stories • Code GUI for sprint 4 user stories • Code functionalities for sprint 4 user stories • Generate Remaining Bug Reports 	
Chong Jun Wei	<ul style="list-style-type: none"> • Create and complete testing for sprint 4 user stories 	
Ivan Lee You Qing		

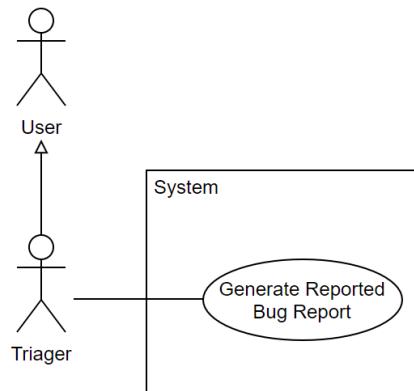
8 Sprint 5

8.1 User Stories and Diagrams

8.1.1 Generate Reported Bug Report (User Story #15)

As a triager, I want to be able to generate a report that shows the total numbers of bugs reported in a selected timeline with all the information for each bug so that I can see the performance of reporters.

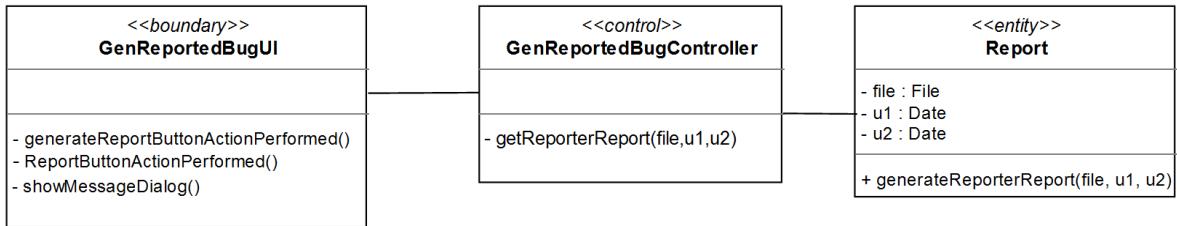
Use Case Diagram



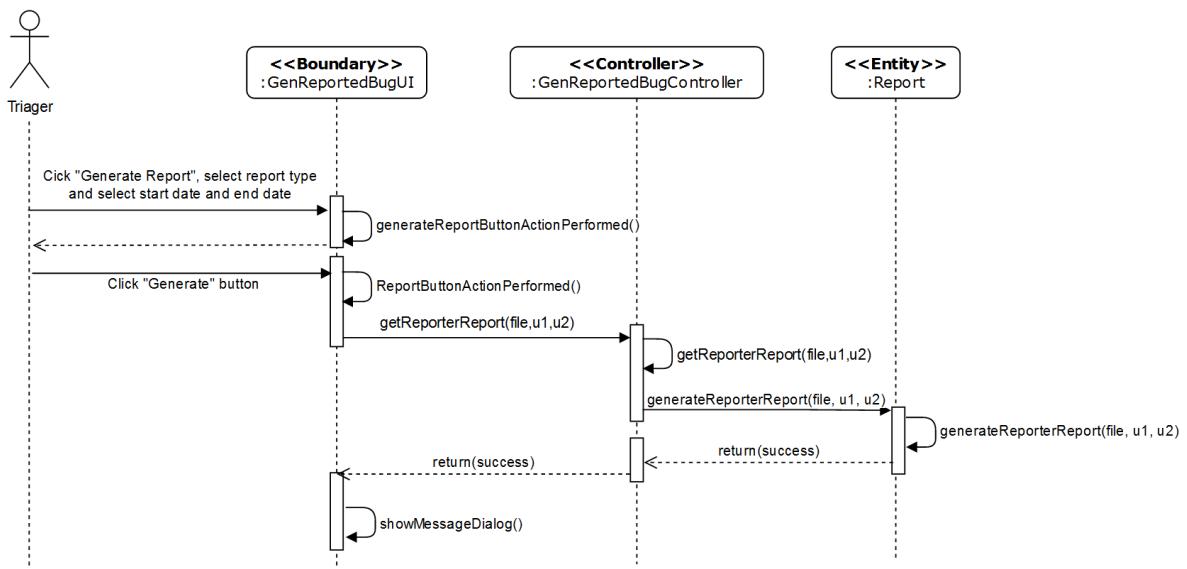
Use Case Description

Use Case Name: Generate Reported Bug Report	ID: 15
Stakeholders and goals: Triager wants to generate a report from the system which shows the number of bugs reported	
Description: A triager wants to generate a report that shows the total number of bugs reported within a selected timeline	
Actors: Triager	
Trigger: A triager in a system clicks on the "Generate Report" button, select "Reporter" for the report type, select a period from start to end and clicks "Generate Report" button	
Preconditions: Triager must be logged in to the system	
Normal flow: <ol style="list-style-type: none">1. Triager clicks on the "Generate Report" button2. System directs to the Generate Report Page3. Triager select "Reporter" from the Report Type drop down list4. Triager select a period from start to end and click the "Generate Report" button5. System will generate and produce all the bugs reported in the selected timeline	
Sub flows: None	
Alternatives/Exceptional flows: <ol style="list-style-type: none">5. Empty report: An empty report will be produced if there are no bug reported on selected timeline	

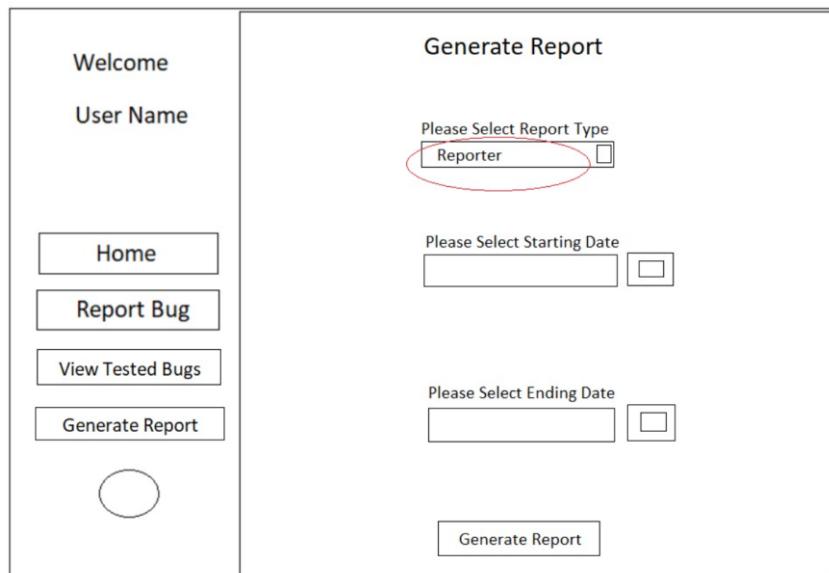
BCE Class Diagram



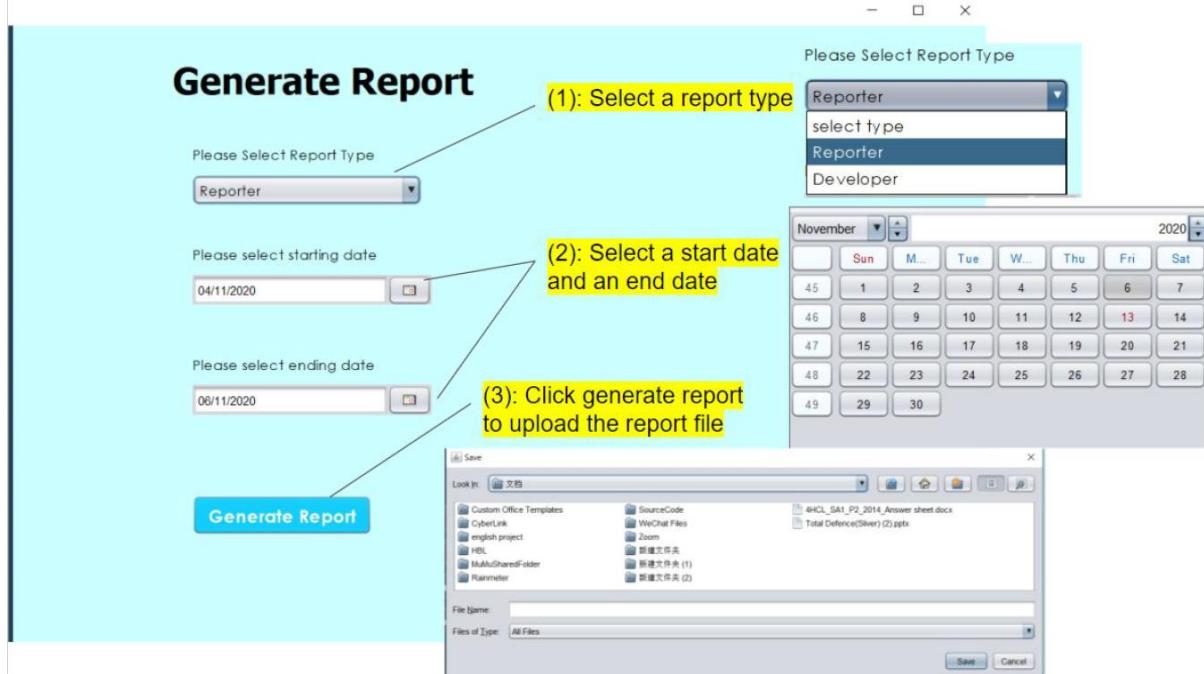
Sequence Diagram



Wireframe



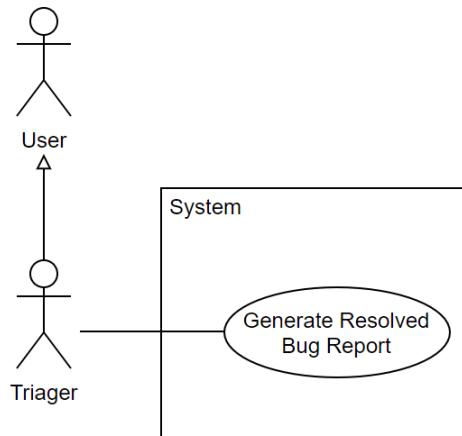
GUI



8.1.2 Generate Resolved Bug Report (User Story #16)

As a triager, I want to be able to generate a report that shows the total numbers of bugs resolved in a selected timeline with all information for each bug so that I can see the performance of developers.

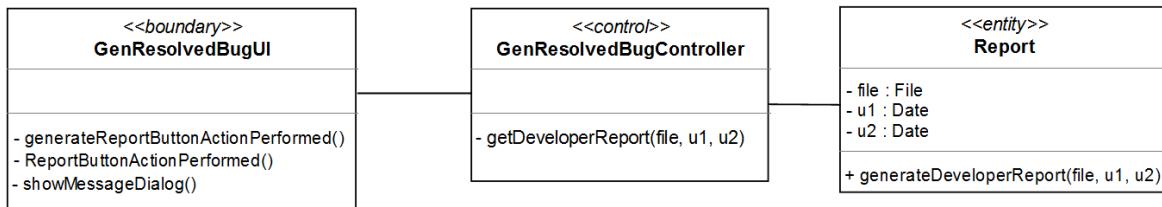
Use Case Diagram



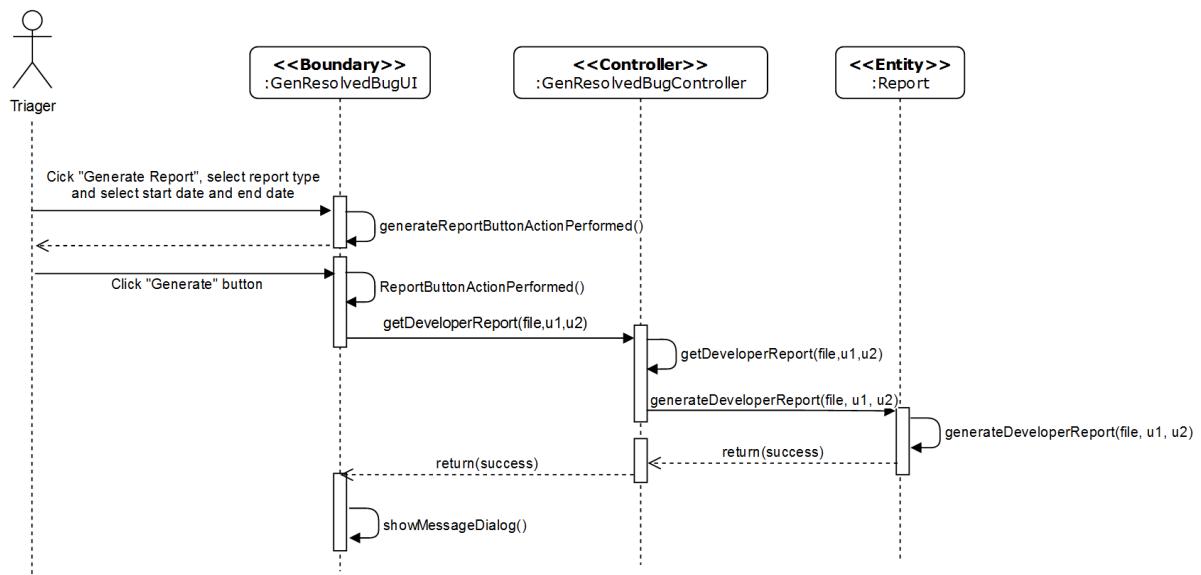
Use Case Description

Use Case Name: Generate Resolved Bug Report	ID: 16
Stakeholders and goals: Triager wants to generate a report from the system which shows the number of resolved bugs	
Description: A triager wants to generate a report that shows the total number of bugs resolved within a selected timeline	
Actors: Traiger	
Trigger: A triager in a system clicks on the "Generate Report" button, select "Developer" for the report type, select a period from start to end and clicks "Generate Report" button	
Preconditions: Triager must be logged in to the system	
Normal flow: <ol style="list-style-type: none">1. Triager clicks on the "Generate Report" button2. System directs to the Generate Report Page3. Triager select "Developer" from the Report Type drop down list4. Triager select a period from start to end and click the "Generate Report" button5. System will generate and produce all the bugs reported in the selected timeline	
Sub flows: None	
Alternatives/Exceptional flows: 5: Empty report: An empty report will be produced if there are no bug reported on selected timeline	

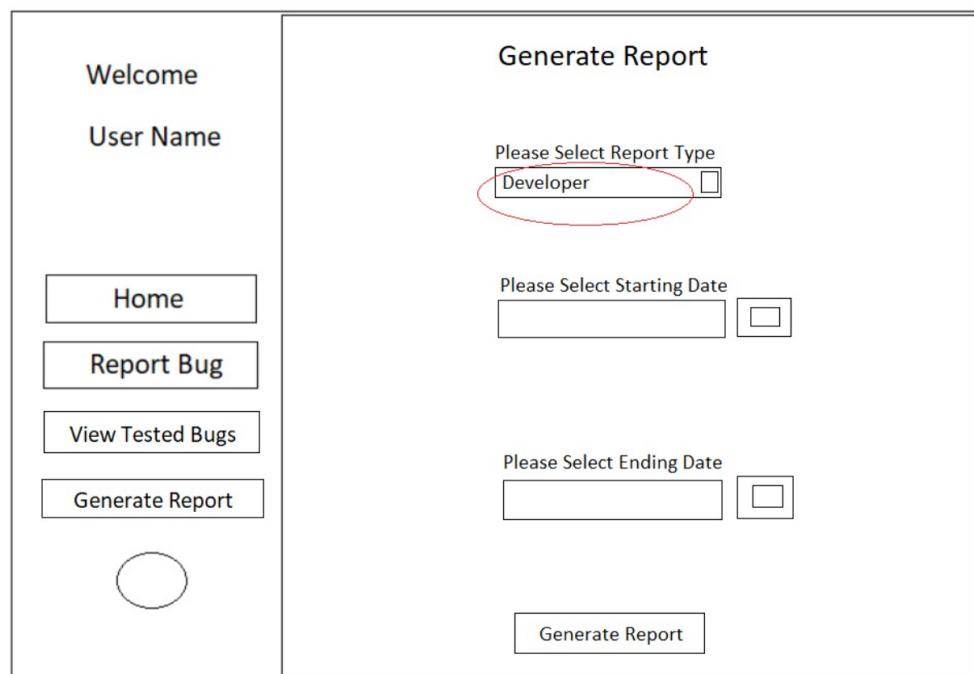
BCE Class Diagram



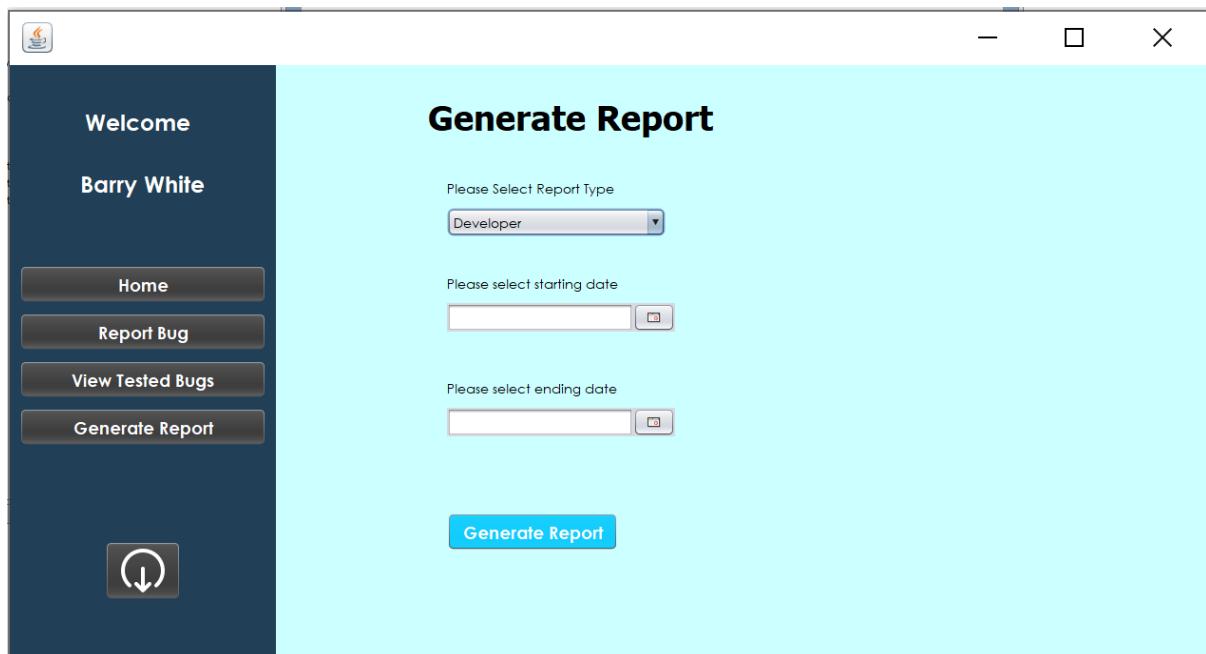
Sequence Diagram



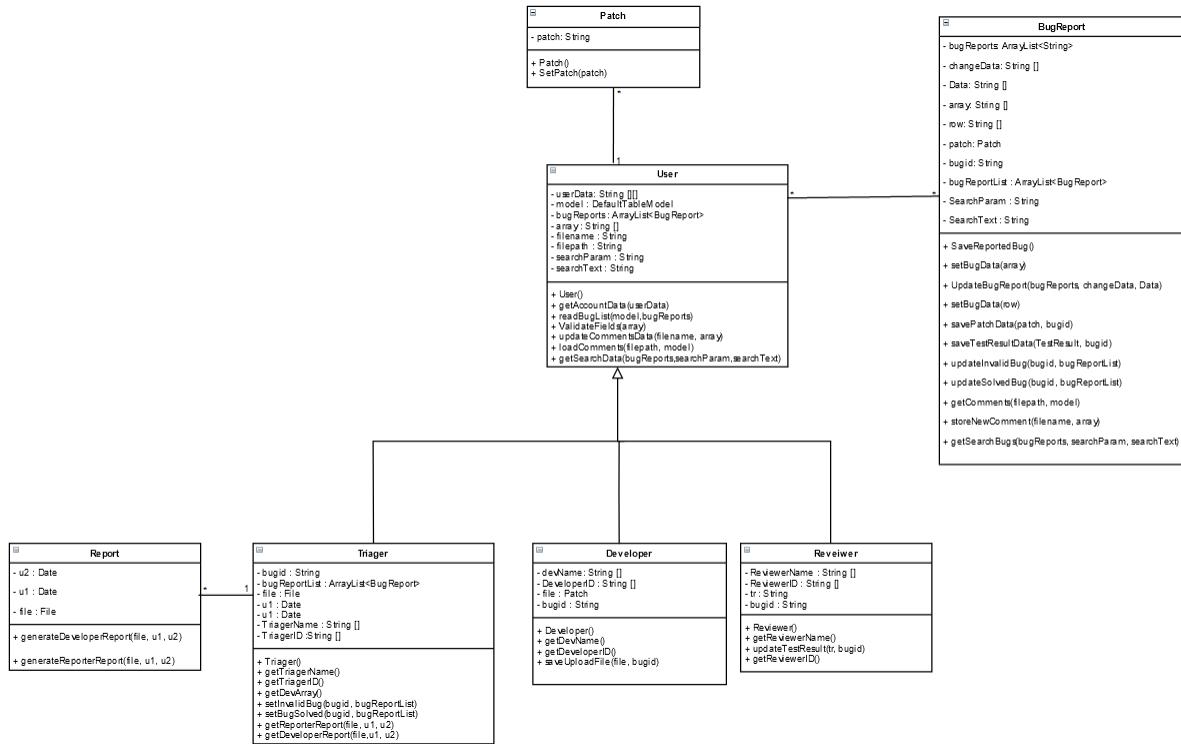
Wireframe



GUI



8.2 Class Diagram



8.3 Test Cases

8.3.1 Functional Test Case

Generate Reported Bug Report

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case	Generate reporter's report						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	3/11/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successfully generating a report to see performance of reporters-Triager	(1) Click on Generate Report Button (2) Select Reporter from the "Please Select Report Type" drop down menu (3) Select a relevant date starting from "Please select starting date" (4) Select a relevant date starting from "Please select ending date" (5) Click on Generate Report Button (6) Input an appropriate name for the report on "File Name" Text box (7) Navigate to the location you want the named report to be saved (8) Click on Save Button	Generate successful message dialog appears on screen saying "Generate Report Successful"	Successful message dialog appears on screen saying ""Named report" successfully created!"	Pass	Dialog box requires you to click ok	

Generate Resolved Bug Report

Project Name:	Bug Tracking System						
Test Case ID:	USER-01						
Test Case	Generate resolved bug report						
Created By:	Jun Wei						
Tested By:	Jun Wei						
Tested On:	3/11/2020						
Preconditions:	Must be logged in						
Test ID	Test Description	Test Data / Step	Expected Results	Actual Result	Pass / Fail	Remarks	Screenshot if fail
USER-01-1	Successfully generating a report to see performance of developers- Triager	(1) Click on Generate Report Button (2) Select Developer from the "Please Select Report Type" drop down menu (3) Select a relevant date starting from "Please select starting date" (4) Select a relevant date starting from "Please select ending date" (5) Click on Generate Report Button (6) Input an appropriate name for the report on "File Name" Text box (7) Navigate to the location you want the named report to be saved (8) Click on Save Button	Generate successful message dialog appears on screen saying "Generate Report Successful"	Successful message dialog appears on screen saying ""Named report" successfully created!"	Pass	Dialog box requires you to click ok	

8.4 Evidence of Use of Methodologies

8.4.1 Meeting Minute

Project Name: Bug tracking system

Meeting Objection: Prepare and begin sprint 5

Date: 29th October 2020

Attendee
Ng Ming Yao
Lim Li Shan
Neo Zhi Kai
Liu Zihan
Mallah Rahul Premchand
Chong Jun Wei
Ivan Lee You Qing

Agenda:

- Define what can be delivered in sprint 5
- Decide on the goal for Sprint 5
- Complete all backlog items
- Work on Sprint 5 Backlog

S/No	Item	Action Item	Due Date
1	Distribution of tasks	Task distributed by MingYao	29 th October 2020
2	Create sprint 5 backlog	Done by Zhi Kai	29 th October 2020
3	Complete all tasks in Sprint 5	Done by team	4 th November 2020

Date and time of next meeting:

Review Final Product: 5th November 2020

Meeting minutes 5 prepared by Zhi Kai

8.4.2 Sprint Planning Meeting

Taiga - Product Backlog (Edited Items)

BACKLOG SIM2020Q4 - TEAM7268

The screenshot shows the Taiga Product Backlog interface. At the top, there is a progress bar indicating 90% completion with 136 defined points, 122 closed points, and 31 points per sprint. Below the bar are buttons for 'SHOW FILTERS', 'SHOW TAGS', 'VELOCITY FORECASTING', and '+ ADD A NEW USER STORY'. A grid displays two user stories:

Votes	User Stories	Status	Points
0	#15 As a triager, I want to be able to generate a report that shows the total numbers of bugs reported in a selected timeline with all the information for each bug so that I can see the performance of reporters.	New	7
0	#16 As a triager, I want to be able to generate a report that shows the total numbers of bugs resolved in a selected timeline with all information for each bug so that I can see the performance of developers.	New	7

8.4.3 Taiga (End of Sprint)

SPRINT 5 (FINAL) SIM2020Q4 - TEAM7...

The screenshot shows the Taiga SPRINT 5 (FINAL) board. At the top, there is a progress bar indicating 100% completion with 14 total points, 14 completed points, 0 open tasks, 18 closed tasks, and 0 pending closes. Below the bar are buttons for 'NEW', 'IN PROGRESS', 'READY FOR TEST', 'CLOSED', and 'NEEDS INFO'. The board has two columns of user stories:

USER STORY	NEW	IN PROGRESS	READY FOR TEST	CLOSED	NEEDS INFO
#15 As a triager, I want to be able to generate a report that shows the total numbers of bugs reported in a selected timeline with all the information for each bug so that I can see the performance of reporters. New 7points				<ul style="list-style-type: none">#33 Draw use case diagram for #15 As a triager#32 Create use case description for #15 As a triager#233 Code GUI for #15 As a triager#235 Testing for #15 As a triager#234 Code functionalities for #15 As a triager#232 Draw Wireframe for #15 As a triager#132 Create Class Diagram for #15 As a triager#131 B-C-E Design for #15 As a triager#132 Create Sequence Diagram for #15 As a triager	
#16 As a triager, I want to be able to generate a report that shows the total numbers of bugs resolved in a selected timeline with all information for each bug so that I can see the performance of developers. New 7points				<ul style="list-style-type: none">#30 Draw use case diagram for #16 As a triager#236 Draw Wireframe for #16 As a triager#239 Testing for #16 As a triager#237 Code GUI for #16 As a triager#238 Code functionalities for #16 As a triager#29 Create use case description for #16 As a triager#133 Create Class Diagram for #16 As a triager#134 B-C-E Design for #16 As a triager#135 Create Sequence Diagram for #16 As a triager	

8.4.4 Sprint Review Meeting

Work accomplished during Sprint 5:

- Revise and enhance existing product backlog items
- Team decided on the goal for sprint 5
- Completed all tasks for items selected from product backlog to sprint 5

8.4.5 Sprint Retrospective Meeting

Work to complete next reporting period:

- Complete User Manual
- Create video for product demo

What went well in the Sprint:

- Most of the tasks allocated to the members can be met at the deadline thus the team is able to continue smoothly within the timeframe

What can be improved:

- Deadlines for tasks must be strict so that there will be no delays while waiting for incomplete work

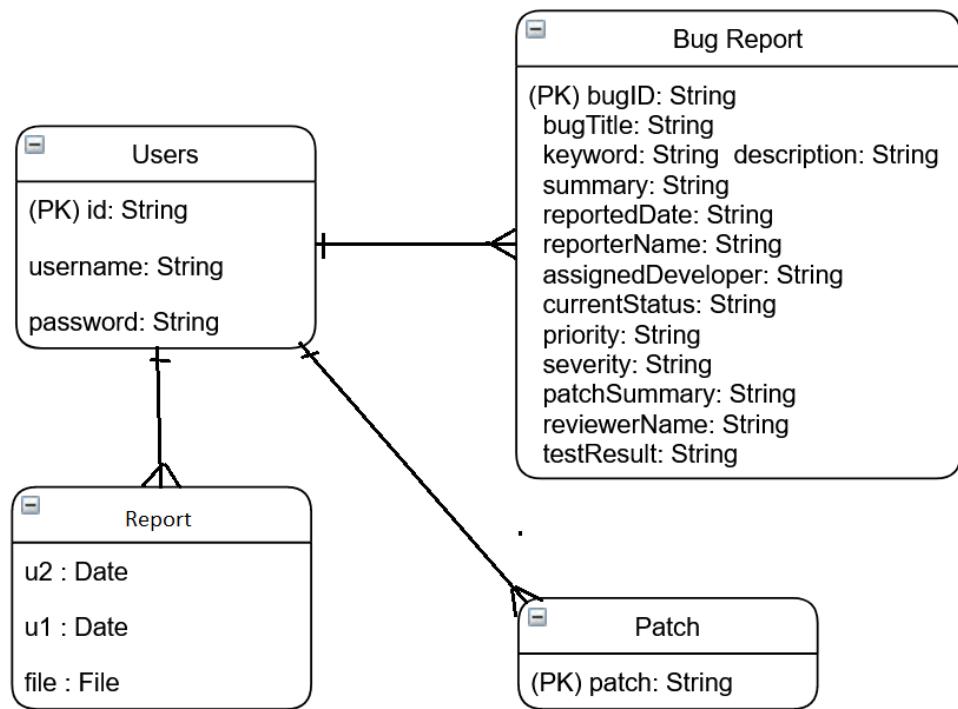
(Discussions)Suggestions/Issues:

- Communication is key, team should check frequently with each other on their progress

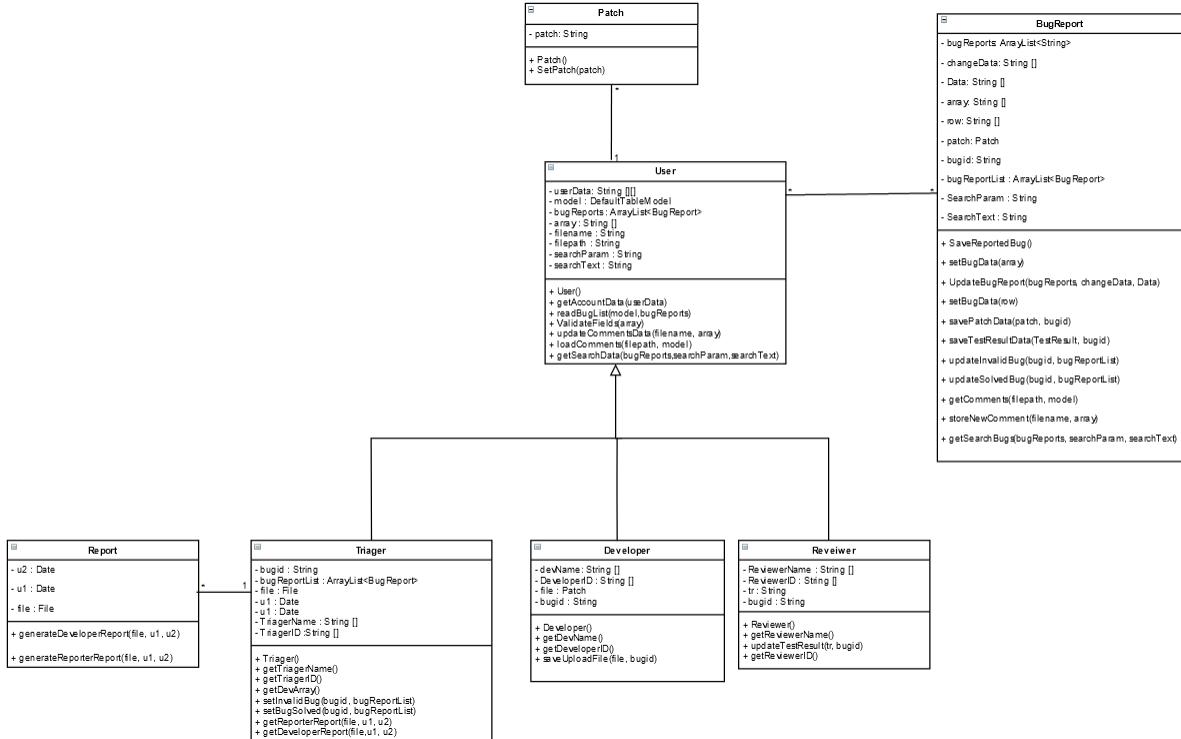
8.4.6 Member Contribution

Name:	Task Completed
Ng Ming Yao	<ul style="list-style-type: none"> • Distribution of roles • Update class diagrams for sprint 5 user stories • Update B-C-E diagrams for sprint 5 user stories • Update sequence diagrams for sprint 5 user stories • Code GUI for sprint 5 user stories • Code functionalities for sprint 5 user stories
Lim Li Shan	<ul style="list-style-type: none"> • Create week 5 report • Create use case diagrams for sprint 5 user stories • Create use case descriptions for sprint 5 user stories • Update class diagrams for sprint 5 user stories • Create B-C-E diagrams for sprint 5 user stories • Create sequence diagrams for sprint 5 user stories
Neo Zhi Kai	<ul style="list-style-type: none"> • Create meeting minutes 5 • Create week 5 report • Create use case diagrams for sprint 5 user stories • Create use case descriptions for sprint 5 user stories • Update class diagrams for sprint 5 user stories • Create B-C-E diagrams for sprint 5 user stories • Create sequence diagrams for sprint 5 user stories • Analyzed and evaluated product backlog • Prepare sprint 5 backlog
Liu Zihan	<ul style="list-style-type: none"> • Create user manual for all sprints • Preparation for demo video • Assist in Testing
Mallah Rahul Premchand	<ul style="list-style-type: none"> • Update class diagrams for sprint 5 user stories • Update B-C-E diagrams for sprint 5 user stories • Update sequence diagrams for sprint 5 user stories • Code GUI for sprint 5 user stories • Code functionalities for sprint 5 user stories
Chong Jun Wei	<ul style="list-style-type: none"> • Create and complete testing for sprint 5 user stories
Ivan Lee You Qing	

9 Data Persistence Diagram



10 Class Diagram



11 Data-Driven Development

In the model requirement stage, we identify that we could use data-driven development to solve two problems in our software. First, implementing machine learning (ML) could help us to solve the problem of user reporting invalid bug report. Secondly, we could use ML to help to identify duplicated bug report. Supervised model could be used in our case as we are able to label our dataset.

Next in the data collection stage, we could use internal data for the training as we have lots of bug reports in our system. In data cleaning stage, we will try to remove those bug reports that might become noise to ML. After removing the noise, we will proceed to label the date record we have and available for ML training. Next, we will prepare for the feature engineering process which will enable us to create a predictive model consists of an outcome variable, which contains data that needs to be predicted.

Now we can finally come to the stage where model training is being carried out. Training on algorithm and tuning would be carry out in this stage to ensure that ML is giving out the output we required. Then we proceed to model evaluation stage to determine is the ML output desirable. Else we might need to repeat the feature engineering stage and model training stage.

Then we will go into deployment after we have achieved the desirable result in model evaluation stage. It would be deployed to our bug report software when data-driven development reaches this stage. Now we will need to continuously monitor the performance and check for any error is made by the AI. Depending on the situation we might need to go back to training stage.

12 Project Closure

BACKLOG SIM2020Q4 - TEAM7268



Final meeting minutes

Project Name: Bug tracking system

Meeting Objection: Preparation for project closure

Date: 5th November 2020

Attendee
Ng Ming Yao
Lim Li Shan
Neo Zhi Kai
Liu Zihan
Mallah Rahul Premchand
Chong Jun Wei
Ivan Lee You Qing

Agenda:

- To Complete all remaining tasks
- To finalize the final deliverables

S/No	Item	Action Item	Due Date
1	Distribution of tasks	Task distributed by Ming Yao	5 th November 2020
2	Complete data-driven development	Done by Ming Yao	15 th November 2020
3	Complete ethical consideration and discussions	Done by Zhi Kai	15 th November 2020

2	Complete final checks on programming codes	Done by Rahul and Ming Yao	18 th November 2020
3	Complete final checks on all documentations	Done by Zhi Kai and Li Shan	18 th November 2020
4	Finish Demo Video	Done by Zihan	18 th November 2020

Final minutes prepared by Zhi Kai