

Development of Defect Tracking System

Syed Muhamad Akif Syed Masohor¹, Norhamreeza Abdul Hamid^{1*}

¹Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: <https://doi.org/10.30880/aitcs.2023.04.01.054>

Received 14 June 2022; Accepted 09 May 2023; Available online 30 June 2023

Abstract: A defect is also known as a bug. The current process of developers and testers from the chosen case study does not use any kind of specific tools to keep track of the list of bugs discovered during or after development. In the current process, the spreadsheet has been used to list defect that has been found and use cloud storage to synchronize with other users. The main purpose of this system is to improve the efficiency of testers and developers to keep track of bugs found during the development process. There are six modules which are the registration and login module, bug submission and display module, communication module, notification module, and report generation module. This system will help developers and testers to report, track and discuss everything regarding the project development more efficiently.

Keywords: Defect Tracking System, Defect Tracker, Bug Tracker, Web-Based, Object-Oriented Approach.

1. Introduction

A defect in computer terminology is an error, faulty instructions, or flaw in a computer program or system that can lead the program to an unexpected behavior which is well known as a bug. The first computer bug that has been discovered in 1945 at Harvard is an actual bug which is a moth trapped between two electrical relays of the Mark II Aiken Relay Calculator that had caused the machine to stop functioning [1]. Today, most of the defects usually happen because of human error while writing the codes for the system or compilation error by the software that has been used to develop the system. Bug usually will be discovered during the development phase of a Software Development Life Cycle (SDLC). Bug reporting is a collection of activities such as investigation, gathering information, testing, and debugging throughout the whole system development process [2]. The defect tracking system is a software application to keep track of a list of reported bugs of a system that will be used by a group of developers and testers. The whole system monitoring will be managed by a project manager. In software development, bug reports provide crucial information to developers [3]. It is unavoidable to have bugs during the development process of a system. If loads of information provided in the bug report are not organized carefully it will cause problems for the developers to work on it [4]. Thus, having a good

defect management tool to keep track of discovered bugs during or after the development of a system will surely help the process of developer to debug the system efficiently.

Kaneka (Malaysia) Sdn. Bhd. is a company that has multiple manufacturing facilities that are located at Kuantan, Pahang. Most of the available machine is being controlled manually or by using a system that is only accessible using internal devices. Due to the pandemic, the company is slowly converting its current system that has been used to control the machine into a web-based application so that it can be easily monitored from afar using other devices. During the development process, every developed system or application will be going through a testing phase.

In the current process, developers and testers from the Information Technology (IT) Department of Kaneka (Malaysia) do not use any kind of specific tools to keep track of the list of bugs discovered during or after development process. The system development team has been using Microsoft Excel to list and track the status of bugs found that can be viewed or edited by everyone who is involved with the project. The spreadsheet file is being shared by using OneDrive to be accessible by multiple people at the same time. The project members have to check the spreadsheet regularly or be notified manually if there are any changes. During the work from home period, most of the communication by email or personal messaging applications such as WhatsApp or Telegram. This situation may also cause confusion if it is not being handled carefully by an experienced project manager or set by specific rules on how to use the spreadsheet. Two developers may have been working on fixing the same bug without even knowing until one or both of them have fixed it. It will be a waste of time when the other developer could have been focusing on other tasks.

Thus, this project proposes a defect tracking system for software development to help a group of developers and testers to manage the list of discovered bugs during or after the development process to keep their workflow become more efficient. Testers can post a ticket on a dashboard for their group with details of the bug and the urgency level to fix it. Developers can view the dashboard to see any available ticket and pick the job if no one has taken it yet and the status of the ticket will be updated as taken by whom. Once the developer has updated it as fixed, testers can review it again to verified it is already fixed before marking it as completed. Therefore, this system is useful for any group of developers to manage their projects during and after development. Bug-tracking helps the software developers in knowing what the error is, resolving it, and learning from it [5].

2. Study of Existing Related System

Based on the study that has been done on the three systems, there are a few common features discovered with different methods of execution. The existing systems are BugZilla, Zoho BugTracker and MantisBT. The Table 1 compare the features between all of the systems including the proposed system.

Table 1: System’s Comparison

Features/System	BugZilla	Zoho BugTracker	MantisBT	Proposed System
System Type	Web-based	Web-based & Mobile	Web-based	Web-based
Log in	Yes	Yes	Yes	Yes
Project Folder	No	Yes	Yes	Yes
Bug Display	Yes	Yes	Yes	Yes
Communication	Yes	Yes	Yes	Yes
Notification	Yes	Yes	No	Yes
Reporting	Yes	Yes	Yes	Yes

After going through the comparison table above, almost all of the main features of the proposed system are available on the compared system. However, some of the features work differently and that makes every system is unique on its own.

3. Methodology

Methodology is the technique that is being used to develop a system. The prototype model has been chosen for the development of the proposed system. As shown in Table 2, each phase of the prototyping methodology has its task and output that need to produce during the entire project development.

Table 2: Software Development Activities and Task

Phase	Task	Output
Planning	– Proposed the project	– Project proposal
	– Determine the project schedule, activities, and output	– Develop Gantt chart
Analysis	– Research related project	– Use case diagram
	– Analyze possible system requirement	– Sequence diagram
		– Activity diagram
		– Class diagram
		– To be model
Design	– Design database for the system	– System architecture
	– Design the user interface	– Database design
		– User Interface design
Implementation	– Program system code	– System prototype
Testing	– Test the system	– Test case result
	– Identify system problem	
System Prototype	– Do detail testing	– Functional system
	– Go back to the analysis phase if necessary	

3.1 System Requirement Analysis

System requirement analysis is the process to decide the requirement that the system needs to have before moving into the development phase. The system requirement will consist of functional and non-functional requirements and user requirements. Table 3 below shows the functional modules for the system.

Table 3: System Functional Module

No.	Module	Function	User
1.	Registration and Login Module	– Manage the user registration and login process	Project Manager, Tester, and Developer
2.	Project Module	– Create a project	Project Manager
		– Add or remove a user in a project	
		– Generate invitation codes for other users	Tester and Developer
		– Join a project	
3.	Bug Submission and Display Module	– Create and update bug submission	Project Manager, Tester, and Developer
		– View list of the reported bug in a table	
		– View details of selected bug submission	
4.	Communication Module	– Discussion between users on the detailed page of a bug report	Project Manager, Tester, and Developer
5.	Notification Module	– Notify every user in the related project when there is a new bug submission	Project Manager, Tester, and Developer
		– Notify every user that is interconnected with the specific bug report when there is an update	
6.	Report Generation Module	– Generate a report of the system usage	Project Manager

3.2 Functional and Non-Functional Requirements

Functional requirements describe the function of the developed system, while a function is described as a behaviour that will change the input to an output. The functional requirements of the proposed system is shown in Table 4 below.

Table 4: Functional Requirements

No.	Module	Description
1.	Registration and Login Module	<ul style="list-style-type: none"> – The system should allow users to register a new account. – The system should notify users if registration is unsuccessful. – The system should be able to verify the users. – The system should notify users if login is unsuccessful. – The system should redirect users to the overview page after login.
2.	Project Module	<ul style="list-style-type: none"> – The system should allow the project manager to create a new project. – The system should allow the project manager to add users to a project. – The system should allow the project manager to remove users from a project. – The system should allow the project manager to view the generated invitation code for a project. – The system should allow a tester or developer to join a project.
3.	Bug Submission and Display Module	<ul style="list-style-type: none"> – The system should allow a tester to create a bug submission. – The system should allow a tester to update bug submission details. – The system should allow a developer to update a bug submission status. – The system should allow all users to view the list of bug submissions. – The system should allow all users to view the details for bug submission.
4.	Communication Module	<ul style="list-style-type: none"> – The system should allow all users to post a comment on a detailed page of a bug submission. – The system should allow all users to view the comments by other users.
5.	Notification Module	<ul style="list-style-type: none"> – The system should notify all users in the project when there is a new bug submission. – The system should notify related users when there is a reply comment. – The system should only notify the related user when a report submission has been updated.
6.	Report Generation Module	<ul style="list-style-type: none"> – The system will only allow the project manager to generate a report. – The system will notify if a report cannot be generated.

3.3 Non-Functional Requirement

Non-functional requirements describe the benchmark that will be used to evaluate the system operation instead of the behaviour of the system function. The non-functional requirement of the proposed system will be shown in Table 5 below.

Table 5: Non-Functional Requirements

No.	Requirement	Description
1.	Operational	<ul style="list-style-type: none"> – The system should be able to access anytime. – The system should be web-based and run on any web browser – The system should run on any modern device with internet access.
2.	Performance	– The system should provide a response time of 3 seconds or lesser.
3.	Security	– The user account should be maintained securely.
4.	Usability	– The system should be user-friendly and easy to use.

User requirements will describe the expectation of the user from the functionality of the system. Table 6 shows the user requirements for the proposed system.

Table 6: User Requirements

No.	User Requirements
1.	All users should be able to login into the system using a valid username and password.
2.	All users should be able to recover a forgotten password using by using email.
3.	All users should be redirected to the respective page after successful login.
4.	The project manager should be able to create a project.
5.	The project manager should be able to add or remove a user from a project.
6.	The project manager should be able to view the generated invitation code for a project.
7.	The tester and developer should be able to join a project.
8.	The tester should be able to create and update a bug submission detail.
9.	The developer should be able to update a bug submission status.
10.	All users should be able to view the list and details of bug submission.
11.	All users should be able to post comments on a detailed bug submission page.
12.	All users should be able to view the comments.
13.	All users within the same project should be notified when there is a new bug submission.
14.	Only related users should be notified of an update of a specific bug submission.
15.	The project manager should be able to generate the report for the system usage.

3.4 Use Case Diagram

For the use case diagram that represents the overall activity of the Defect Tracking System for Kaneka (Malaysia). The purpose of the use case diagram will illustrate the parts of the system. Refer to Appendix A.

The use case diagram consists of registration, login, creating a project, joining a project, creating or viewing bug submission, updating bug submission details or status, posting or viewing comments, generating reports, and receiving notifications.

3.5 Class Diagram

A class diagram is a graphical notation that has been used to create and visualize an object-oriented system. The objective of this project is to develop this system using an object-oriented approach. Therefore, a class diagram will be used to model the class of the proposed system. Refer to Appendix B for the class diagram.

For the user class, it has the id, password, name, email, mobile, and type. The project class, has the id, title, description, invite code, and date created. The report class has id, title, priority, description, status, submitted by, assignee, date created, date modified, and project id. The comment class has the id, author, content, date created, parent id, and report id. Lastly, the notification class has the id, title, description, status, project id, report id, and user id.

3.6 To-Be Model

The proposed system is expected to be completely developed by the end of this project. The system should be able to improve the efficiency of the tester and developer to develop their system, especially for debugging purposes. The whole development team will have all of the information needed and have a discussion about any defects in their system in a single medium. Refer to Appendix C for the to-be model of how the proposed system is going to work

3.7 System Design

System design is the designing process for the elements of the system which includes system architecture, module and component, database design, and interface design. It is the process of decomposing a system into many components to understand how the component interacts. The goal of

system design is to provide sufficient components for the implementation that will satisfy the requirements [6].

3.7.1 System Architecture

The system architecture is a conceptual model that describes the whole structure, behavior, and system views. It includes each small component of the system and how it will interact with each other. Refer Appendix D for the system architecture of the proposed system.

3.8 Database Design

The database design is a process collection that simplifies the process of designing, developing, implementing and maintenance of a system data management. It is an instrument that is being used to store information or data [7]. By designing a database, it can determine what data need to be stored, what data type for each data and how will the data will relate to each other.

3.8.1 Schema

Schema is an outline, diagram, or model for the system database. It will describe the tables and the data in the database. The database schema for the proposed system is listed below:

- i. Account(id, password, name, email, mobile, type)
- ii. Project(id, title, description, invite_code, date_created)
- iii. ProjectMembers(project_id, user_id, date_joined)
- iv. Report(id, title, date_created, date_modified, priority, description, submitted_by, assignee, status, project_id)
- v. Comments(id, author, content, date_created, parent_id, report_id)
- vi. Notifications(id, title, description, status, project_id, report_id, user_id)

4. Result and Discussion

This section will discuss the result and test that has been conducted for the proposed system.

4.1 Implementation

The proposed system is being developed by using HTML5, CSS, and JavaScript for the front-end, and core PHP Programming Language for the server-side scripting. MySQL is being used as the database for the whole system. During development, Visual Studio Code is the software that is used to write the source code and Laragon is the software that includes Apache and MySQL for local hosting.

4.1.1 Registration and Login Module

This module allows users to register a new account or log in using their existing accounts to gain access to the system. If the user logged in by using an invalid account or the wrong password, the system will show an error message to alert the user about the problem. Figure 1 shows the login page of the system and Figure 2 shows the registration page of the system

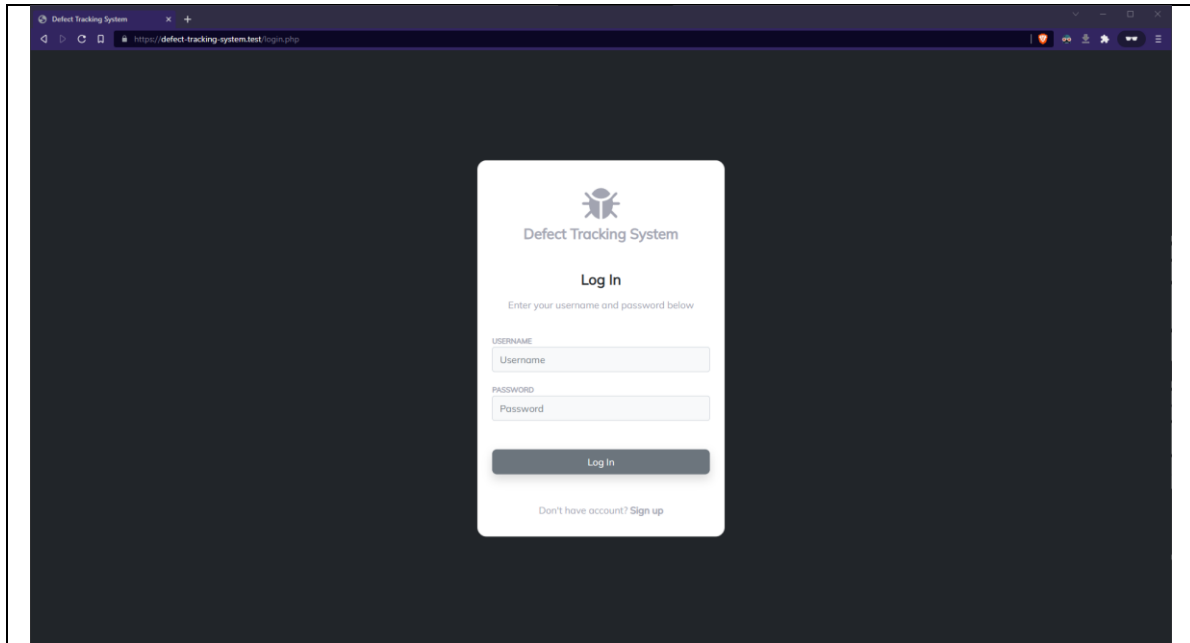


Figure 1: Login page of the system

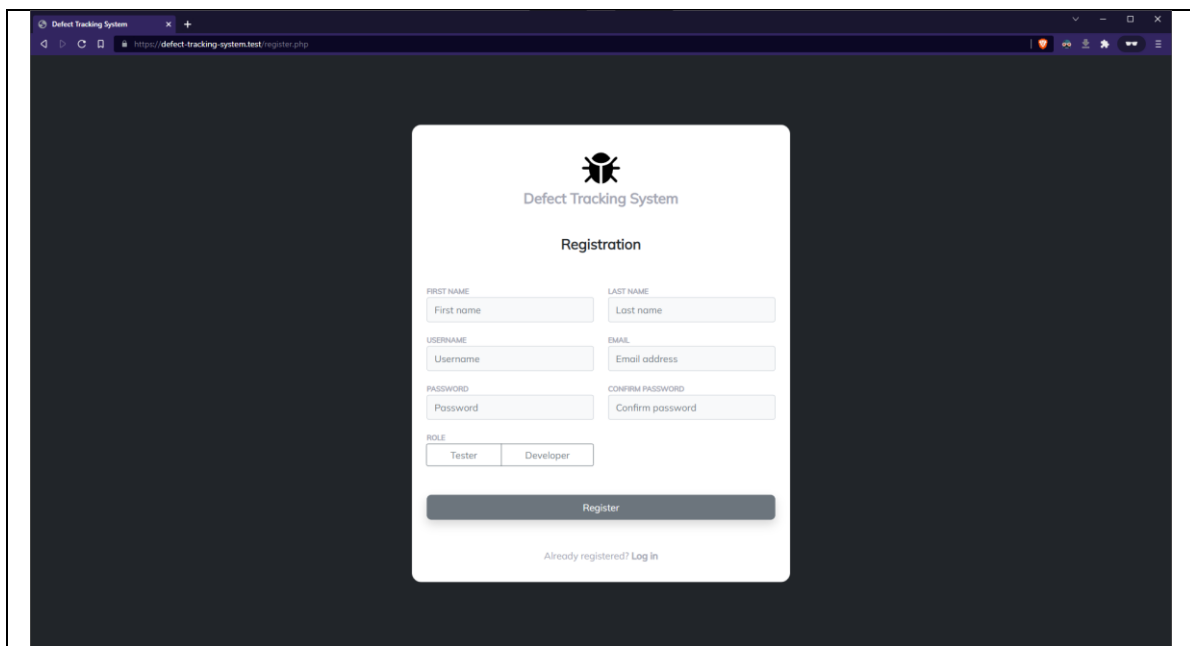


Figure 2: Registration page of the system

For new users, a new account can be created by using the registration page that can be accessed through the sign-up button. If the user enters an invalid email, the system will alert the user and show an example of a valid email address. The system will alert the user if the entered password does not match or the password is too short before submitting the registration form and it will also alert the user if the username is already being taken.

4.1.2 Project Module

This module will be focused on user involvement with the projects. Depending on the user type this module will have some different features. The project manager will be able to create a new project, add or remove and invite members. The project manager will be required to fill in the project title and

description before submitting the form. Figure 3 shows the project creation modal and Figure 4 shows the add member modal.

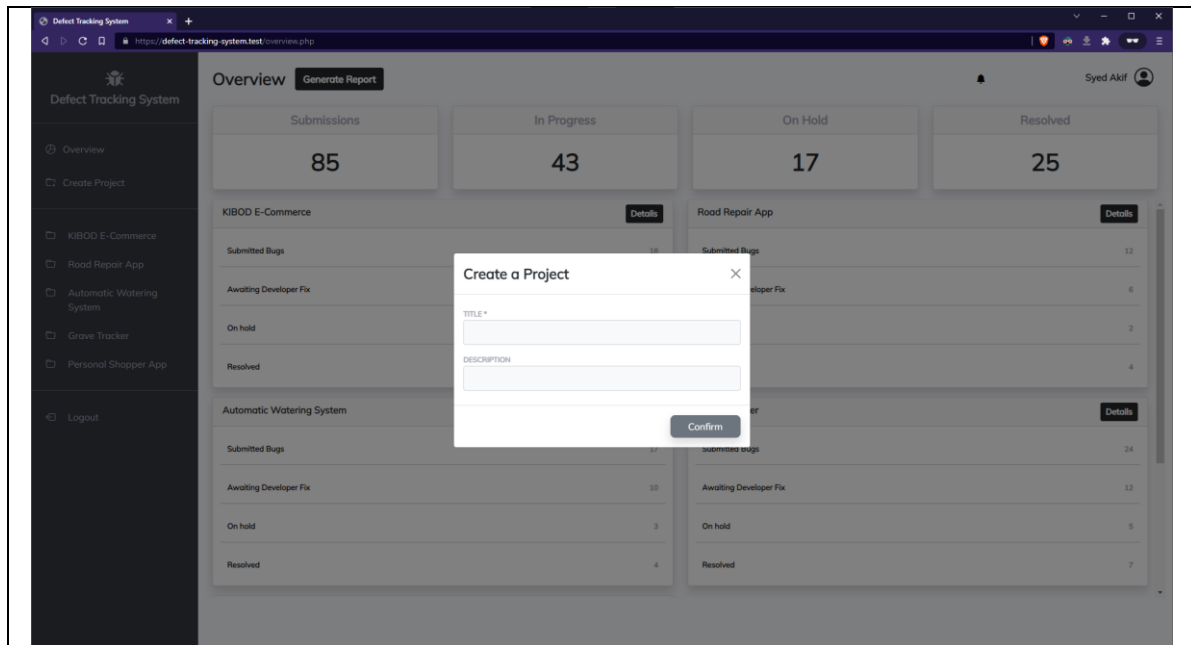


Figure 3: Create project modal

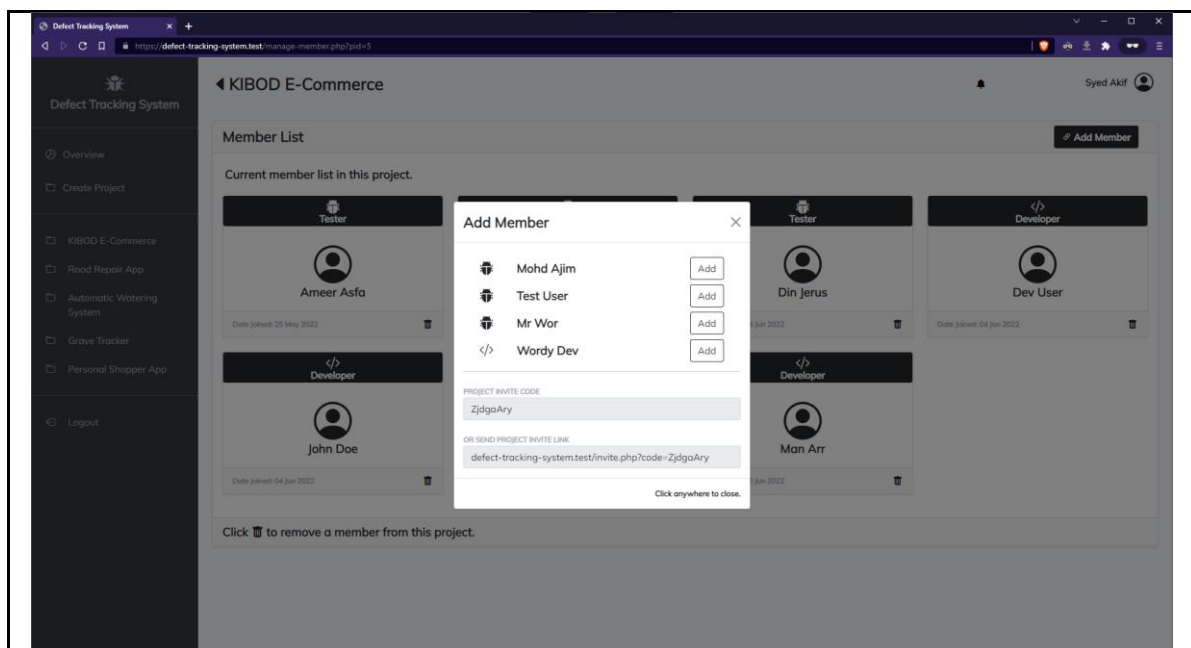


Figure 4: Add member modal

Once created the system will generate a unique invite code for others to join. Other users can join a project based on the invitation code or link that has been provided by the project manager.

4.1.3 Bug Submission and Display Module

This module will allow the reporter to submit a report and the system will display all of the submission lists in table format. Figure 5 shows the list of submission and Figure 6 shows the new submission form.

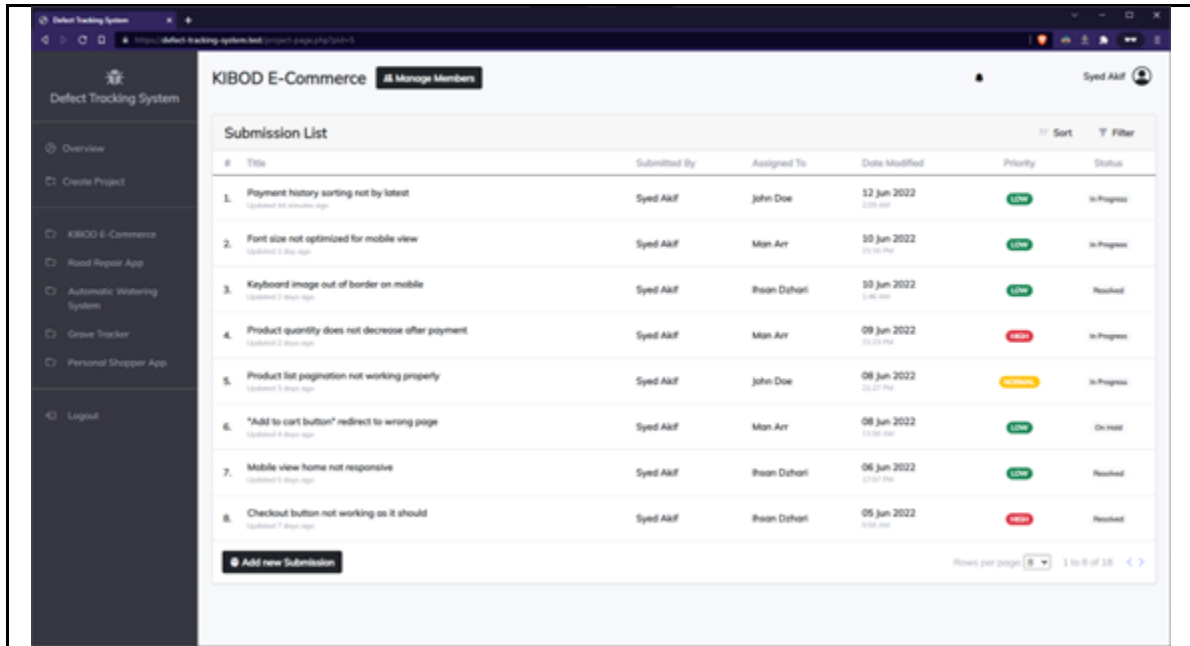


Figure 5: List of submission

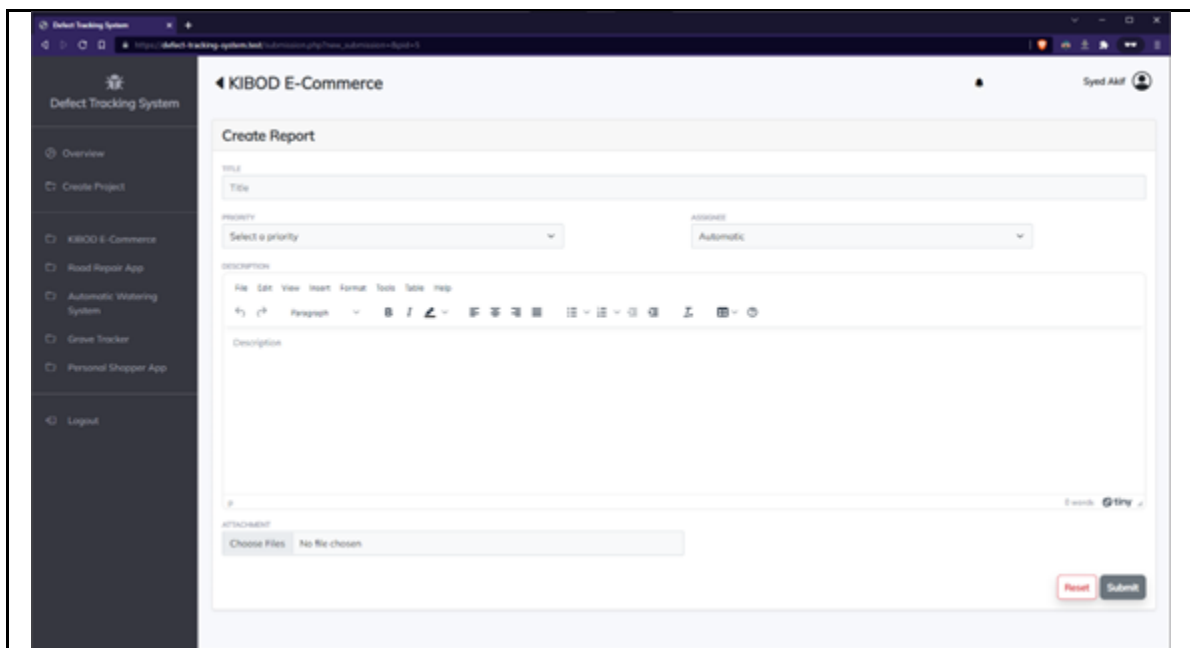


Figure 6: New submission form

To submit a report, a tester will be required to fill all of the input fields, select the priority of the submission and select who it will be assigned. Details for each submission can be viewed by clicking on the title that will be displayed on the table on the project page.

4.1.4 Communication Module

This module will allow any user to have a discussion among the project members on any of the selected reports. Users can write and post a comment that can also be viewed or reply by other users. Figure 7 shows the comment section on the detailed submission page.



Figure 7: Comment section on the detailed submission page

4.1.5 Notification Module

This module will send a notification to every user when a new submission has been added to any of the projects that are already being joined. It will also notify the specific user when any changes on the related submission.

4.1.6 Report Generation Module

This module is only available to the project manager which gives access to generate a report that consists of data from the usage of the system.

4.2 Testing

The testing phase will begin once the system prototype has been developed. This phase is one of the most important parts of the whole development phase to ensure the quality of the prototype. The test cases for every use case are shown in Table 8 while the overall result is in Table 7 below.

Table 7: Overall Result Test Case

Test Case Module	Passed Test Case	Rate (%)
Registration (TEST_100)	5/5	100
Login (TEST_200)	4/4	100
Project (TEST_300)	6/6	100
Bug Submission and Display (TEST_400)	8/8	100
Communication (TEST_500)	4/4	100
Notification (TEST_600)	2/2	100
Report Generation (TEST_700)	2/2	100
Total	31/31	100

Table 8: List of Test Cases

Test Case	Software Requirement	Description	Result
TEST_100_01	SRS_REQ_101	The system loads the registration page view	PASS
TEST_100_02	SRS_REQ_102	New users register with input personal data	PASS
TEST_100_03	SRS_REQ_103	The minimum password length should be 8 characters	PASS
TEST_100_04	SRS_REQ_104	The system displays an error message if the registration is not successful	PASS
TEST_100_05	SRS_REQ_105	The system stores user information in the database once registration successful	PASS

Table 8: Cont'

TEST_200_01	SRS_REQ_201	The system loads login page view	PASS
TEST_200_02	SRS_REQ_202	Users log in to the system with valid information	PASS
TEST_200_03	SRS_REQ_203	The system displays an error message if the login is invalid	PASS
TEST_200_04	SRS_REQ_204	The system redirects to the correct page based on the user account	PASS
TEST_300_01	SRS_REQ_301	The system only allows the project manager to create a project	PASS
TEST_300_02	SRS_REQ_302	The system only allows the project manager to add a user to a project	PASS
TEST_300_03	SRS_REQ_303	The system only allows the project manager to remove a user from a project	PASS
TEST_300_04	SRS_REQ_304	The system only allows the project manager to view the invitation code for a project	PASS
TEST_300_05	SRS_REQ_305	The system should only allow users to join a project with a valid invitation code	PASS
TEST_300_06	SRS_REQ_306	The system displays an error message if the join process is unsuccessful	PASS
TEST_400_01	SRS_REQ_401	The systems load bug submission form	PASS
TEST_400_02	SRS_REQ_402	Users create a bug submission	PASS
TEST_400_03	SRS_REQ_403	The system displays an error message if the submission is not successful	PASS
TEST_400_04	SRS_REQ_404	The system stores report information in the database once the submission successful	PASS
TEST_400_05	SRS_REQ_405	Users update a bug submission	PASS
TEST_400_06	SRS_REQ_406	The system displays the list of bug submissions from the database	PASS
TEST_400_07	SRS_REQ_407	Users view the list of bug submission	PASS
TEST_400_08	SRS_REQ_408	Users view the details for each bug submission	PASS
TEST_500_01	SRS_REQ_501	Users post a comment on selected bug submission	PASS
TEST_500_02	SRS_REQ_502	Users reply to a comment	PASS
TEST_500_03	SRS_REQ_503	The system stores comment in the database	PASS
TEST_500_04	SRS_REQ_504	Users view past comments	PASS
TEST_600_01	SRS_REQ_601	Users receive a notification for a new bug submission	PASS
TEST_600_02	SRS_REQ_602	Users receive a notification only on related bug submission	PASS
TEST_700_01	SRS_REQ_701	The system only allows the project manager to generate a report	PASS
TEST_700_02	SRS_REQ_702	The system displays an error message if the report generation is unsuccessful	PASS

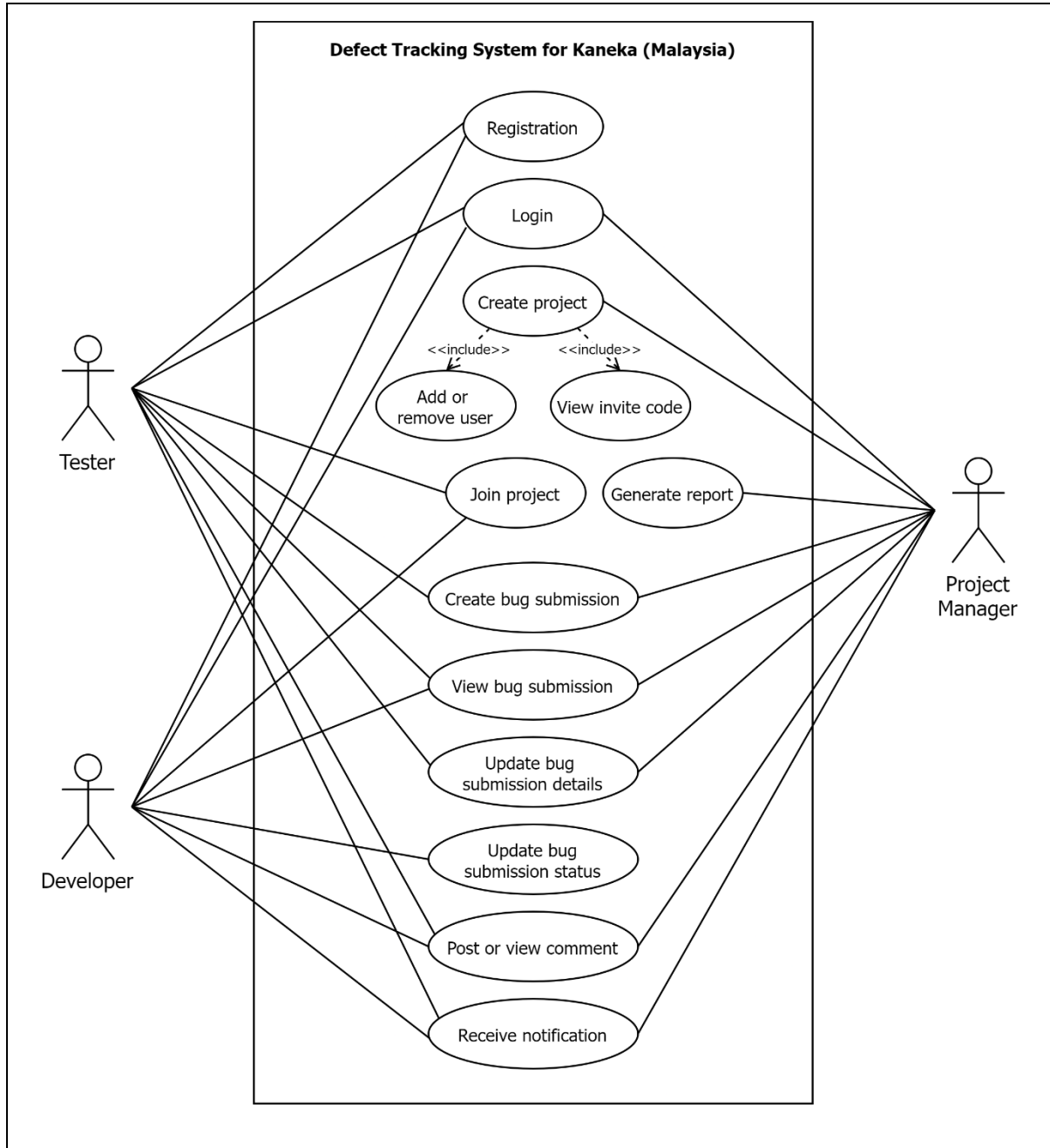
5. Conclusion

The defect tracking system can help a bunch of developers and testers to communicate more efficiently on solving the discovered bugs. The tester will be able to generate a report and record in the system dashboard the bugs that have been found and the developer will be able to work on the specific problem based on what has been reported. Once complete, the developer will update the report and it will proceed to the review process to make sure it is completely fixed. The project manager will be able to monitor the status of every bug discovered in the system. By using the defect tracking system, the developers and testers can report and solve any reported bugs more efficiently.

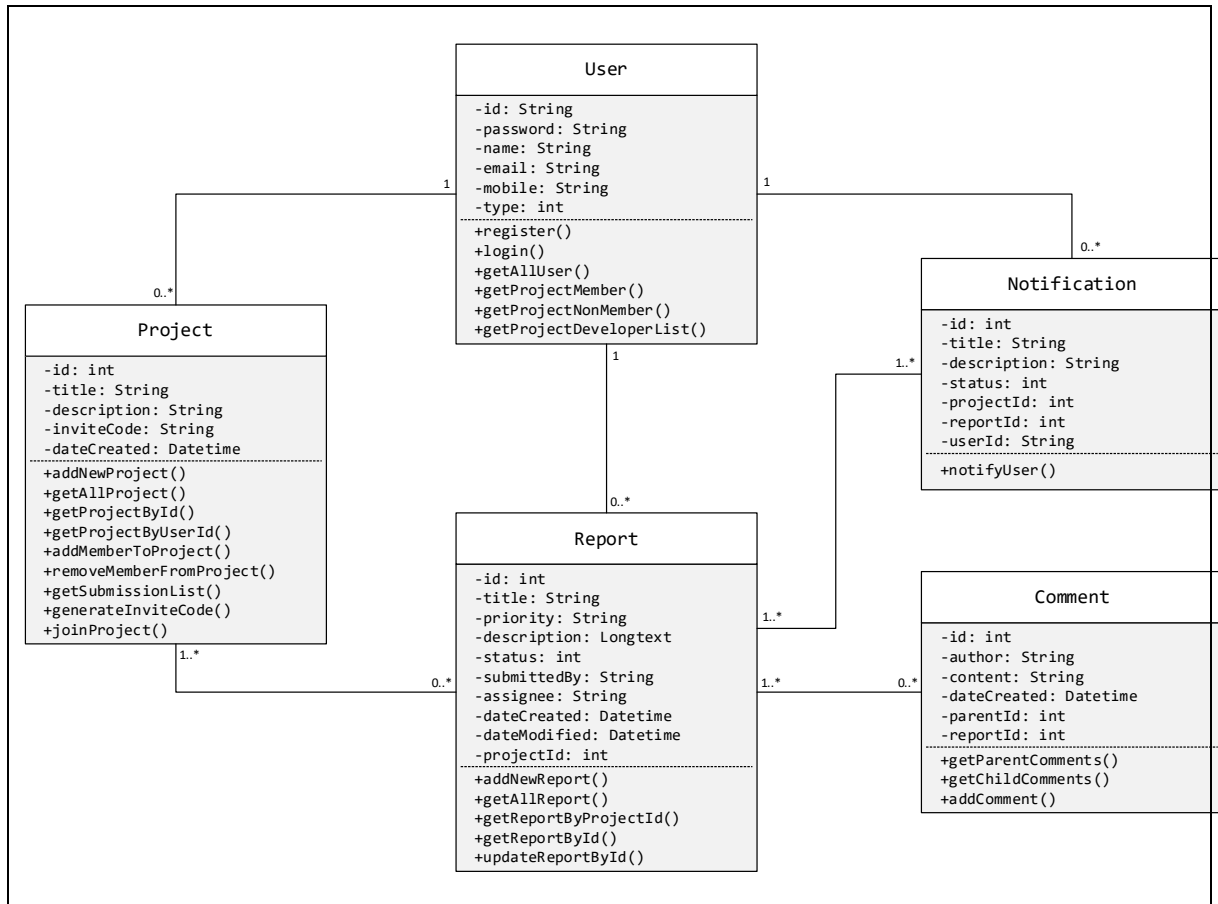
Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

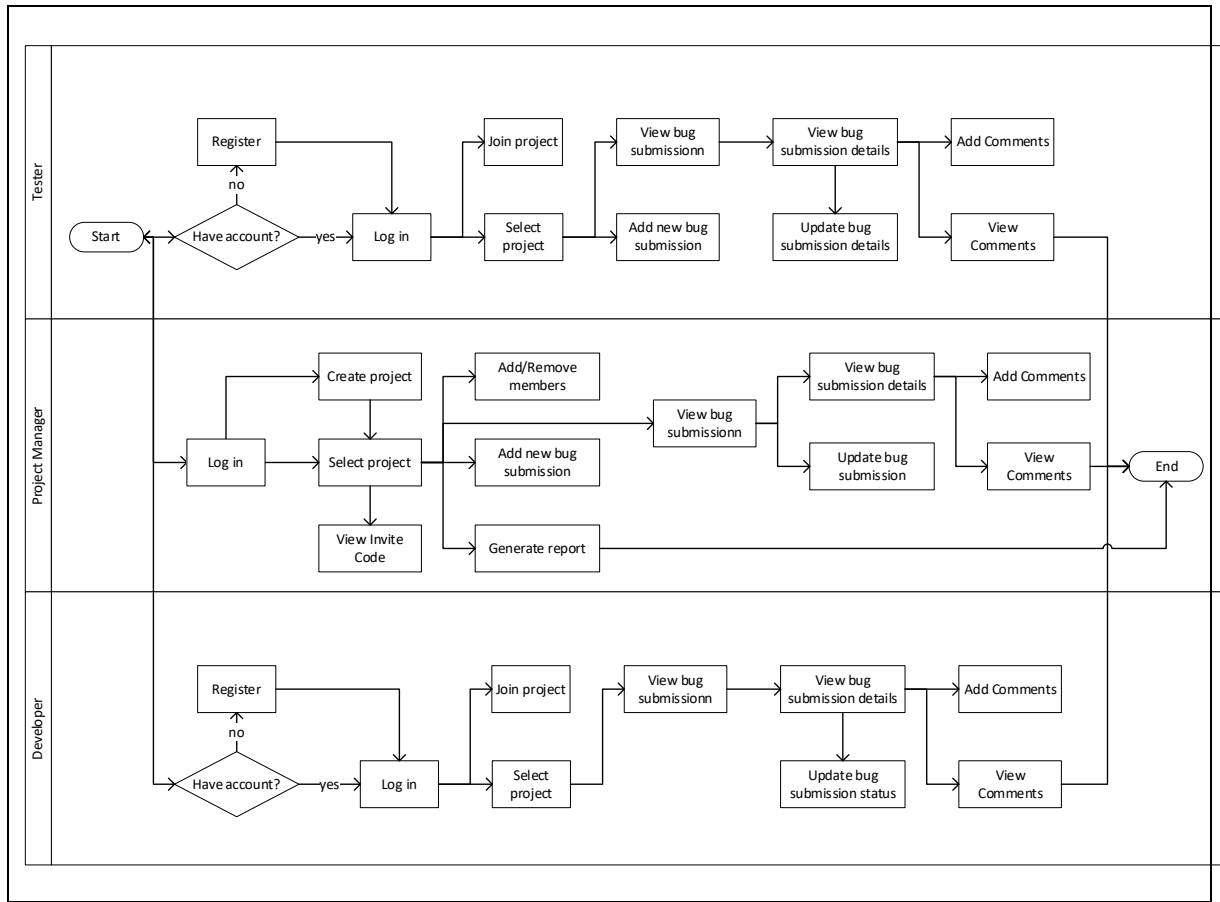
Appendix A



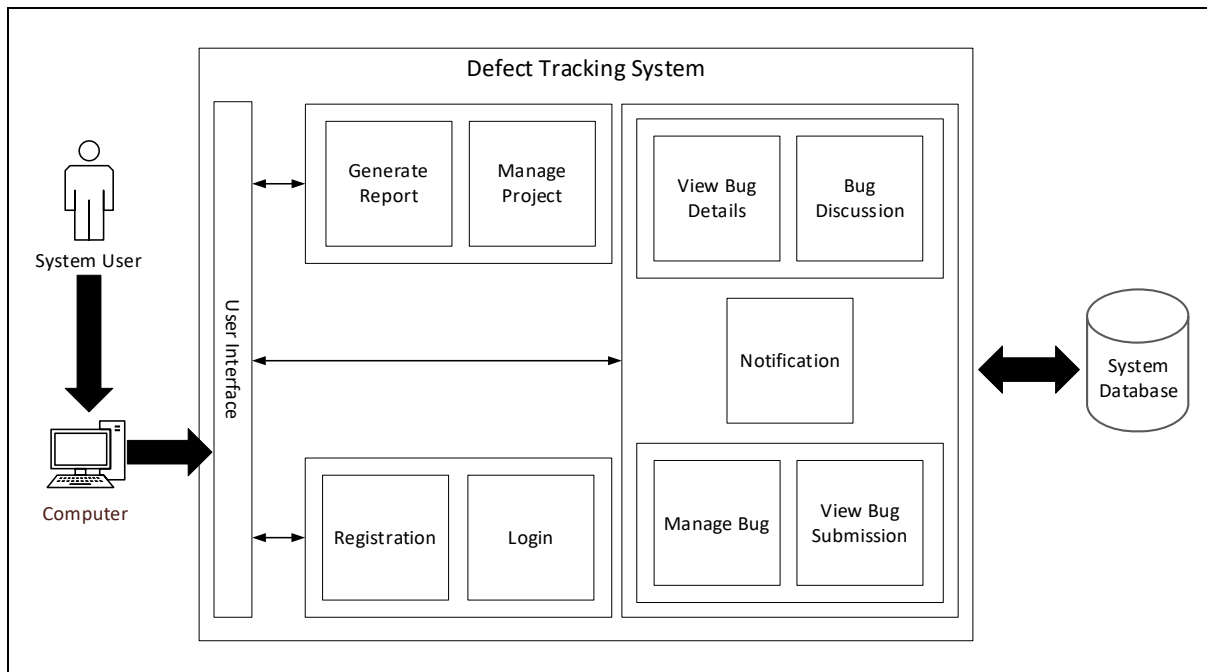
Appendix B



Appendix C



Appendix D



References

- [1] Kumar, R., Gattaiah, D. Y., Shahi, S., & Nagendra, T. A. (2013). Improving software quality assurance using bug tracking system. *International Journal of Computer Science and Information Technologies*, 4(3), 492-497.
- [2] Singh, S. (2013). *Analysis of bug tracking tools*. *International Journal of Scientific & Engineering Research*, 4(7), 134-140.
- [3] Zimmermann, T., Premraj, R., Bettenburg, N., Just, S., Schroter, A., & Weiss, C. (2010). *What makes a good bug report?*. *IEEE Transactions on Software Engineering*, 36(5), 618-643.
- [4] Kolluri, A. B., Tameezuddin, K., & Gudikandula, K. (2012) Effective Bug Tracking Systems: Theories and Implementation. *IOSR Journal of Computer Engineering (IOSRJCE)*. ISSN, 2278-0661.
- [5] Serrano, N., & Ciordia, I. (2005). Bugzilla, ITracker, and other bug trackers. *IEEE software*, 22(2), 11-13.
- [6] Braude, E. J., & Bernstein, M. E. (2016). *Software engineering: modern approaches*. Waveland Press.
- [7] Lerdorf, R., Tatroe, K., Kaehms, B., & McGredy, R. (2002). *Programming Php*. " O'Reilly Media, Inc."