

Change and Bug Tracking System: Anjung Penchala Sdn. Bhd.

Zatul Amilah Shaffiei
Faculty of Computer Science and
Mathematics
Universiti Teknologi MARA
40150 Shah Alam, Selangor
Malaysia

Mudiana Mokhsin
Faculty of Computer Science and
Mathematics
Universiti Teknologi MARA
40150 Shah Alam, Selangor
Malaysia

Saidatul Rahah Hamidi
Faculty of Computer Science and
Mathematics
Universiti Teknologi MARA
40150 Shah Alam, Selangor
Malaysia

ABSTRACT

Defects or bugs have been existed as a problem in the system and they are normally inevitable in software development. Most software bugs arise from mistakes and errors made by people in either a program's source code or its technical design. Huge amount of bugs could be found in system development. It was relatively difficult to manage bugs in simple word documents or remember everything in one's head. Thus Change and Bug Tracking (CBT) System has been introduced in order to keep track of the reported bug in the system. By using this tracking system, every user can report and follow up the issue in the system. This system provided a platform for users from different departments to report issues and understand the latest progress in a product. It consolidated and organized information in single system and centralized data repository for the changes and bugs in system. Besides, this system could be accessed by team members concurrently to improve efficiency and they can search particular information in the system easily. It made communication between teams more effective and all the bugs and changes are recorded in web based system and hence the system is scalable.

General Terms

Security, system development.

Keywords

Bug tracking, web-based application, centralized data repository

1 INTRODUCTION

Bug tracking is a process of keeping track of reported software bug in products. It allows developers to further refine software design by making continual changes or upgrades to the product in order to better serve the customer base. Change and Bug tracking system is a web-based application that keeps track of the change request and bugs found in a system. Once added, each item is tracked through to completion with useful report displays to show progress. People need to keep track all of the issues because they tend to forget and miss the issues that found in the system [1].

Target users for CBT would be clients, software testers, project team members, management members and software developers. It enables multiple users to track and manage multiple projects simultaneously. This system is taking care of the development and support teams as well. According to the research by [1], when the developers have finished a task, they can proceed to work on the next issues while the Quality Assurance (QA) team will start testing. The QA team can use the bug tracking system

to record issues progress, until such time that a developer could buy time to look into the other issues. If the issue is critical, developers can attend to that right away. However, if the issues are not showstoppers, the team of developers can choose to fix them according to the priorities of the activities on which they are working now.

Therefore, it is not only tracking the issues, but it also allows QA team to interact with developers more effectively. Each reported bug has to be tested and reproduced by QA team. They have to simulate the scenario and produce the test results. This system allows user to insert attachments such as MS word and images in the comment so that developers can get the results or useful information faster and then take appropriate action on it immediately.

Keeping good records of decisions made will help to control the scope of the project. Bug tracking also enables you to know what was done with identified issues, why the system was changed, and what future changes to consider. This application will be written using PHP, JavaScript and MySQL and it is implemented as web application, access through intranet or extranet (Internet) using web browser. It supports multiple users and information can be accessed at the same time by multiple users from any type of computer without installing software on user's computer. Since it is a web-based application, users can collaborate from anywhere as long as they have Internet access and a web browser.

This project proposed an enhancement for Change and Bug Tracking System for the use of IT Department employees in Anjung Penchala Sdn Bhd, Petaling Jaya, Malaysia.

1.1 Problem Statement

1.1.1 Defects in application

When an issue is reported, it might involve many tasks such as investigation, information gathering, testing and debugging throughout the whole process. It is rather inconvenience to manage issues of a project for tracking every finding in documents because hundreds of bugs can be found. [2] pointed out those developers create the bugs, but QA team is the one who examine the code and application to discover the exact sequence or combination of steps that turns up an error.

1.1.2 Communication between users

System QA team and system developers will need more effective communication. Research by [3] stated that trying to keep all these bugs in your head or in a single document such as spreadsheets courts immediate disaster because you will not be

able to communicate effectively with one another or with the development team and thus will not be able to contribute to increase product quality. As a project grows, the first problem that is likely to be encountered is that only one person can modify the spreadsheet at a time [4]. According to [1] he also agrees that communication is a critical factor for the success of a software project. Regardless how technically competent the team members are, if the communication is not efficient, the entire team could fail.

1.1.3 Complexity

There are many similar bug tracking systems available for this purpose; however, some of them are created with unnecessary complex and confusing functions. The flow is not straight forward is a main concern. Most bug tracking systems are built in web-based application. Web-based applications can often be difficult to navigate, especially if they behave more like a sequence of HTML pages than an application. Some Web applications are constantly reloading pages because responding to user input requires a round trip between the Web server and the user's browser. It incurs slowness as involving many reloading of pages and makes the application difficult to work with.

1.2 Objectives of Research

This research study has three main objectives that have been accomplished at the end of the research. The objectives are:

1.2.1 To centralize collection of information

This system provides a platform for users from different departments to report issues and understand the latest progress in a product. It consolidates and organizes information in single system and centralized data repository for changes and bugs in system. Without a proper system, project stakeholders cannot access to those information easily.

1.2.2 To provide better communication channel

It makes communication between teams more effective. It keeps all information that related to an issue in a single system and records users' feedback in different categories. Based on [1] it supports the developers, QA team, and users can collaborate with each other through the bug tracking system. They can educate each other on the nature of the issue and on the potential solution.

1.2.3 To simplify the process of bug tracking

This system is focused on simple concepts and usability to provide an easy to use solution for small and medium businesses. The system provides user friendly interface and easy to use functions for tracking issue. As a result, it simplifies every step of tracking issues, thus it can save time and work can be done efficiently.

1.3 Significance of Research

The main idea of this project is to provide an easy-to-use application, simplified change and bug tracking system. The issue reporting process is relatively easy and simple. Users will not feel burden to use this system to track issues as it minimizes the steps and provides clean navigations throughout the system.

In addition, for security feature, only administrator is able to add new user, change user account status and change the user's password. The purpose of this design is to make sure that only administrator can manage and control the system. Normal user

will not have the privileges to do that. Thus it is secure in protecting the password.

2 LITERATURE REVIEW

2.1 Issues and Challenges in Bug Tracking System

Software development project faces multitude of challenges. One of those challenges is the defects found in the project. Defects or bugs are problems exist in the system and they are normally unavoidable in software development. Use of the term "bug" to describe inexplicable defects has been a part of engineering jargon for many decades and predates computers and computer software. A software bug is the common term used to describe an error, flaw, mistake, failure, or fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unintended ways. Most bugs arise from mistakes and errors made by people in either a program's source code or its technical design.

According to [3] the present of bugs has caused the system fail to meet customer's reasonable expectation of quality. Quality is defined as meeting the customer's requirements and suitable for the use of users. Defect or bug is something that does not allow the product to meet customer requirements, or maybe does not fulfill the aspect of fitness for use. The quality problem causes the customers' dissatisfactions or project failure. [5] stated that defect is not an accident, but it occurs because something has not worked as planned. Since that, people realized that implementing bug tracking process is important in order to find, identify and remove bugs or issues within application and eventually keep the system running properly. Besides, the purpose of tracking bug is to improve the quality of software products; it is an eternal truth that newly written software packages will contain bugs and it is inevitable. When a system gets a bug, it can cause all sorts of problems. Bugs can cause the system to perform task incorrectly or cause things to occur that should not be. This can be especially annoying to users who may be unable to reach certain areas of the system or to perform certain operations on the system due to a bug in the system. Once a software product is released to customers, all organizations should adhere to some process of managing customers problems for the ultimate purpose of customer satisfaction.

A helpful tool is needed to ease the situation stated above. Development team needs a systematic way which can help in tracking bugs with all information about them; such system is known as bug tracking system [6]. Development team has to solve the bugs found in order to make the system works in proper way. However huge amount of bugs can be found in a system development life cycle and bug tracking become challenging.

According to the article by [7], it is more challenging to start tracking bugs during the software development phases because software developers are not very cooperative in tracking down their faults. They feel that it threatens both their creativity and professional ego.

Previously, people tracked bug in a simple way. Some users reported bugs by sending email to related technical department. This is rather difficult to keep track the bug reported as the emails are scattered around email application. While developers

of programs with modest popularity can accomplish manual bug tracking via email, it would be a logistical nightmare for developers of more popular software to attempt this approach. In this case, the simplest structured way to keep track of a small number of bugs is to use a spreadsheet program, such as Excel [4]. The main advantage of this approach is its simplicity and suitable for small-scale project. Spreadsheets are commonly available tools and are reasonably well documented and users found it easier to use because of its features. Besides, basic reports are also easy enough to produce in spreadsheet. However, the disadvantages of this approach are significant. Most of these are the same reasons why people use databases rather than spreadsheets. As project grows, the first problem that is likely to be encountered is that only one person can modify the spreadsheet at a time, users cannot modify the spreadsheet concurrently and it is not efficient. One has to wait until other user has finished viewing or editing the spreadsheet and then only able to modify it. Another problem is that, unlike with a good database, if the computer or server running the spreadsheet crashes during modification of bug, then the data can be inconsistent.

Besides, when the total number of bugs in the spreadsheet grows into thousands, the time taken for the different reports generating can grow significantly. Therefore, using spreadsheet for keeping track of bugs should be viewed as a temporary measure because it will not scale as a project grows. The truth is that it is relatively difficult to manage bugs in simple word documents, spreadsheet or remember everything in one's head when it involves big scale of project. The information tracked in this way is difficult to maintain. Eventually the history of the debugging process or valuable information might be loss. Still, a spreadsheet is better than an email folder of reports of possible bugs or a wall full of sticky notes, but it is not scalable and not recommended for large scale project. Due to this reason, bugs tracking system is created to help QA team and development team to keep track of the reported bugs. There are many simple, low-cost bug tracking systems available that will scale better than a spreadsheet.

2.2 Trends in Bug Tracking System

Bug tracking database has been introduced to replace spreadsheet. In recent years, bug tracking systems are widely deployed in the information technology industry, especially in software development. Software and hardware companies use bug tracking systems during the development cycle to track bugs and design issues. Some use it during later phases of the product lifecycle to keep track of defect reports and to obtain quality indicators. The advantage is that this system contains database that stores all issues reported by users. Previously, many older bug or defect tracking systems are client server based and it caused inconveniences to users at the same time. Users need to install the server, and each user need to install the client software. If external users were involved, it could be problematic because of issues like firewall blocking. Also, it is not always feasible to install client software in every user's computer.

Nowadays newer systems are more likely developed in web browser based and thus no client software installation is needed. A web-based bug tracking system is especially attractive if users involved are located in different locations and are connected through the internet. With the aid of internet access, user can

access real-time information from anywhere in the world, 24 hours a day, 365 days a year. A bug tracking system allows users to report a detected bug directly to the concerned people who can fix it. The system maintains a database that records all facts about all defected and reported bugs. The details about a bug may include its severity, the unexpected program behavior, the identity of the user who reported it and identity of expert who is fixing it. Having a bug tracking system is extremely valuable in software development, and they are used extensively by companies developing software products.

All in all, to track bugs, many organizations still rely on Word documents and Excel spreadsheets, but these tactics are inefficient and error-prone to say the least. A good automated issue-tracking solution should streamline the process of raising, managing and fixing issues because bug tracking system should be useful software which can be designed to help software developers in keeping track of reported bug in their other software products. Besides, this can be seen as one of the form of issue tracking system.

Actually bug tracking is a misleading term because it is depending on what kinds of issues need to be tracked. [8] agreed that the system can be called as issue tracking as issue is a broad term to describe most of the kinds of tasks might need to track when developing an application. The issue tracked might include bugs and feature-changes instead of tracking bugs or defects only. As it evolved along the time, this system is not just designed for internal teams such as developers and testers, but also for clients or other end users to report issues related to the system. This system enabled clear visibility of the software quality, and improved the planning and estimating the research and development maintenance effort internally.

2.3 Research on Other Similar Current Systems

Many companies use bug tracking system in developing software products because it is extremely useful and valuable. Research by [9] supports that bug tracking system is needed to store all related data together because most teams do not adequately maintain their requirements and design documents throughout the life of project. Bug tracking system provides important benefits to software development. According to the research by [3], he stated that a bug tracking database facilitates clear communication about bugs. Well-written, standardized reports tell the story much better than free form emails. However, bug tracking could be tedious in the way of doing it. [3] mentioned that it takes significant effort and discipline to use bug tracking system as some might think it is too hassle.

There are numerous bug tracking tools available both commercially and as open source. Select the best bug tracking tool that fulfill the users' requirement is not simple. There are many factors that need to be considered as follow:

2.3.1 Application setup

Application setup is the first step to be done before the system can be utilized. Bugzilla is probably the most well-known of the open source issue-management tools. It is used on many open source projects such as Mozilla, Eclipse, and many Linux distributions, and is well-adapted to large, open projects. But sometimes freeware bug tracking tools might be difficult for user to configure. [10] also agreed that freeware bug tracking

tool such as Bugzilla usually takes a long time to setup or maybe not user friendly and supported. The setup process is not easy and complicated. Therefore users have difficulty to install the application in server.

According to [11], Bugzilla is a tool with an uninspiring user interface and is rich in features, but undeniably cumbersome to install and to maintain. Besides, [12] also supported that Bugzilla is easy to use and quite flexible, but it is very difficult for a novice to install and maintain. Although step-by-step directions have been provided, Bugzilla is not always easy to get working. It is frustrated if users take longer time and lots of work to setup the application in order to make it working. The setup process should be simple and straightforward.

2.3.2 Reporting process

The bug reporting process should be simple and yet useful in tracking bugs. Therefore the steps should be simplified and maximizing communication effectiveness between users. If the reporting process is complicated and not user friendly, users may hassle about using it and it will be time consuming. Since web-based bug tracking system is widely used nowadays, it incurs some problem at the same time. Web-based applications can often be difficult to navigate, especially if they behave more like a sequence of HTML pages than an application. For example, some web applications are constantly reloading pages when responding to user input requires a round trip between the Web server and the user's browser. Eventually the process might be cumbersome and users might later refuse to utilize it to track the issues related to system. The complexity of the system might reduce the efficiency of work and users might feel troublesome when use the system.

Some bug tracking systems are available with some rarely used features which increase the complexity of the reporting process. As a result, the process of reporting bugs or issues into a system must be easy and straightforward. Bug tracking system can be complicated or simple. Ideally it should be simple and fast so as to avoid people resorting to email or other text document. For example, a bug tracking system called, "Bug Genie", it is a powerful bug tracking system with many features. But the process of reporting bug involves many steps and not straight forward. Feedback from users who used this system stated that configuring projects or adding new issues to the tracker has been made an overly difficult task and it is overly complicated bug reporting.

According to [13], if the bug tracking system is difficult to use, is time-consuming, or asks a lot of information that the author of an incident report sees no need for, use of the system will be limited or the data may be not accurate. Therefore an effective bug tracking systems can improve customer satisfaction, raise productivity and reduce downtime.

2.3.3 Notification method

Besides that, notification is an important feature in bug tracking system as users may not logged into system all the time. Notifications are really a workflow feature because they help you track and manage bugs. For example, notifications can keep team members informed about important changes to bugs, such as changes in priority or addition of newly added information. Notifications can also help automate the defect management process. For example, system can notify development managers when new bugs are submitted and developers when bugs are

assigned to them. Most systems such as "Flyspray" and "BugTracker.NET" use email only to notify users when there are updates in the system but there is a possibility that emails might be blocked by spam filter and not delivered to the designated recipients. In this case, email notification might not be a reliable channel to notify users.

In Web 2.0, Really Simple Syndication (RSS) feeds provide another way to keep users up-to-date. Users require RSS reader to receive incoming feeds and the feeds will not be blocked by spam filter. According to [14], RSS is a delivery channel to provide content to subscribers and it can increase awareness of users about updated issues in the system. If RSS is used properly, users can quickly and easily get the updated content immediately without having to stay in the system all the time. Therefore bug tracking system should include RSS feeds as one of delivery methods so that system can notify users in various ways instead of email only.

3 SYSTEM ANALYSIS

In system analysis phase, the researcher used several fact finding techniques to gather information about the bug tracking system. These techniques help the researcher to study and analysis the bug tracking system. Fact-finding is an important activity in system investigation. In this stage, the functioning of the system is to be understood in order to design the proposed system. Various methods are applied for this and these fact-finding techniques are described as below:

- Document review - The researcher has reviewed the document include bug reports, system guideline of the system being used in Anjung Penchala Sdn Bhd. This company is using an open source bug tracking system, which is called **Bug Genie**.
- On-site observation - The observation of current operating procedures had been done in Anjung Penchala Sdn Bhd. First-hand knowledge of the activities, operations, processes of the system on-site was obtained.
- Internet research - The major source of the Internet research is Universiti Teknologi MARA (UiTM) digital library and Google Books as they offer great volume of books, journals and articles that related information.
- Interview – interviews technique had been done with the potential users to collect information about the current system. The areas of misunderstanding, unrealistic exception and descriptions of activities and problems were discovered along with resistance to the new proposed system.

4 RESULTS AND FINDINGS

4.1 System Development Environment

Both hardware and software are needed in order to build an application system. It will speed up the development process if appropriate hardware and software are used in system development. The following tables show the hardware and software required in order to develop and run the system.

Table 1. Hardware and Software Requirement

Hardware	Description
Processor	AMD Phenom II X2 550 Processor 3.10 GHz
Memory	4.00 GB
Hard Disk space	1TB
Monitor	21.5 Samsung Sync Master 2233sw
Others	Other required standard computer peripherals, such as keyboard and mouse.

Software	Description
Server Environment	
Operating System	Microsoft Windows 7
Programming Tool	Adobe Dreamweaver CS4
Database server	MySQL 5.1.30 database (Windows)
Web server	Apache 2.2.11
Client Environment	
Internet browser	Internet Explorer 6.0 or above, Google Chrome, Mozilla Firefox

4.2 Issue Tracking Life Cycle

This Issue Tracking Life Cycle is an adopted from Bugzilla Bug Life Cycle [15]. The system design had been proposed according to the user requirements and the detailed analysis of a new system. Each issue in the system will have an associated status, priority, and severity. The status indicates the current progress with respect to an issue.

Table 2. Issue status type and description

Issue Status	Description
Open	It is available for someone to be picked up and solved it
In-Progress	Someone is working on the issue, so no one else should worry about picking it up
Resolved	Someone provided a solution to the issue
Closed	Someone verified the solution and the issue is no more

The set of states forms the life cycle of an issue. Refer to Figure 1 for the life cycle of an issue in the issue tracking system. When a bug is found, an issue is created in the issue tracking system, and the issue will be in Open state. Once a developer starts working on the issue, the issue will transit to In-progress state. When the developer provides a fix, the issue is Resolved. Then, the fix needs to be verified. If the verification fails, the issue will be re-opened and will go back to the Open state. If the verification is passed, the issue can be closed, which indicates that the bug is fixed. Priority indicates how soon the issue needs to be attended to. High priority issues are to be attended before the others. Severity indicates the gravity of the issue. In other words, severity indicates the impact of the issue on the system.

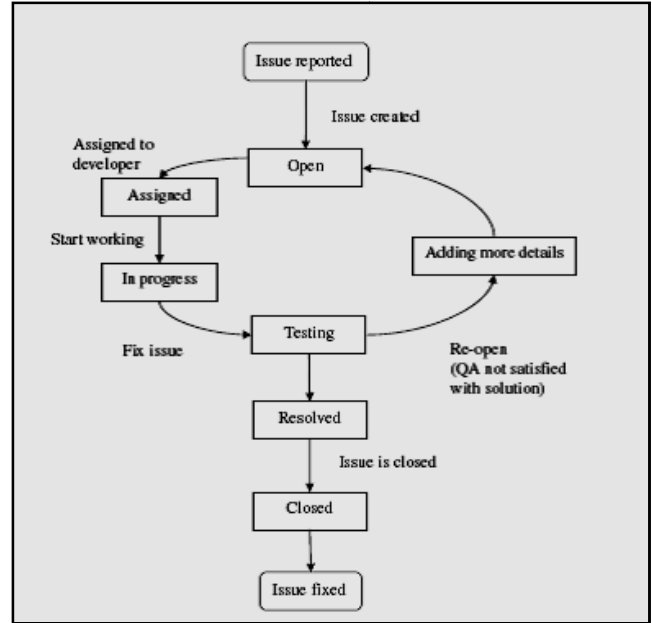


Figure 1. Issue Tracking Life Cycle

Figure 2 shows the overall flow in CBT System which involved five subroutines; form data validation, member page, forgot password process, issue track and profile update.

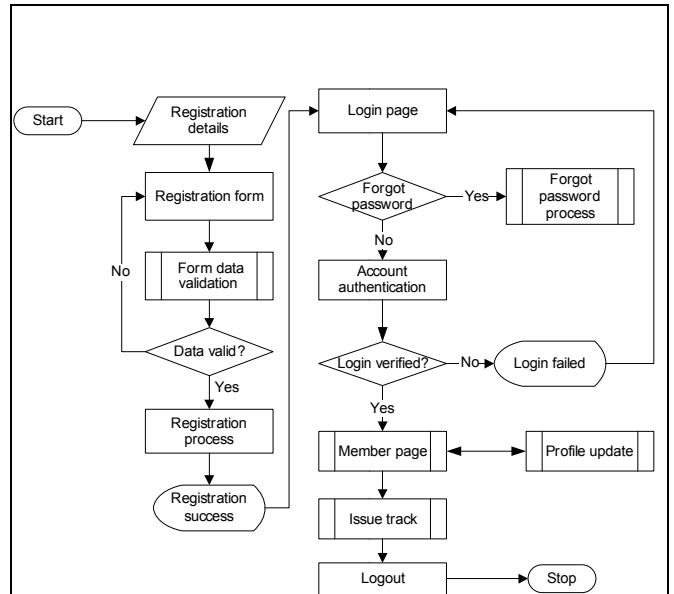


Figure 2. Overall flowchart of Change and Bug Tracking System

4.3 CBT System Architecture

As a result, the CBT system had been proposed to be used by every departments in Anjung Penchala Sdn Bhd. Technical team will be the main user. Their responsibility is to complete and update issues arise. Meanwhile the head of IT department will be the system admin that shall maintain and manage overall system flow. Basically they have different roles in using the

system. All users will be connected to database using user interfaces. These interfaces are based on five main modules:

- Project management - This module creates and maintains products in the system. Multiple products can be created and users can report issue in products accordingly. Each issue reported in the system has its status to indicate the progress. User can define milestones and set important dates on project timeline by assigning issues to milestones. User can manage the issue details such as status, remark, history, attachment and comment. System keeps issue history that showing changes have been made to an issue over its lifetime and users can view the history that attached in the issue.
- User management - This module is used to create and manage users in the system. Administrator is able to create unlimited members with user profiles, store passwords encrypted in the database and restrict projects to certain groups of users. Users created are categorized as clients, software testers, project team members, management members and software developers. Each user category has different access rights in the system as they take different roles and responsibilities in the software development progress.
- Email management - Users with specified access right can determine the email recipients and assign selected email addresses to products accordingly. Auto email notification will be generated and sent to related users based on the email recipients list when issue is reported.
- Dashboard – This is the doorway page after users logged into system. Overview is provided to let users know what are the highlighted issue, importance notice (posted by other users), upcoming due date in a project. Besides, Really Simple Syndication (RSS) is available and provides feeds for reported issue and updates in the system.
- Web calendar - It allows user to add tasks in a specified dates. After task is added, the selected date shows the title of the task, user can click on the date to view and edit the task.

4.4 Program Architecture

CBT system was developed by using PHP (Hypertext Preprocessor) as server-side scripting language. PHP was chosen because it is suitable for dynamic web sites development and it is the widely-used, free to download and use, yet efficient alternative to competitors such as Microsoft's ASP. Besides, it is one of the most popular open source languages used in web development and it can be downloaded from the official PHP resources: www.php.net. It is flexible, fast, and simple in its requirements, yet powerful in its output [16].

As for the client-site scripting language, Javascript had been used which is an integral language of Web application development. It can be used to create an interactive web site. Other than that, in the development of CBT system, jQuery from JavaScript library has been used in several areas, such as data searching, data sorting, table manipulation and date picker. It is

simple and easy to integrate jQuery to the web site and perform desired functionalities. It is a fast, concise, library that simplifies how to traverse HTML documents, handle events, perform animations, and add AJAX. It provides a wide range of features, an easy-to-learn syntax, and robust cross-platform compatibility in a single compact file [17].

4.5 Function Architecture

The functionalities of CBT are categorized in the following modules such as project management, user management, email management, dashboard and web calendar. Table below shows the level of data retrieval between the user and CBT system.

Table 3. Function Architecture with CBT system User

Function\Type of User	New User	Member	System Admin
Project Management			√
- Add new project			
- Update project status		√	
- Submit new issue		√	
- Update issue			
User Management	√		√
- Register new user	√		
- Update user details			
- Change password			
- Reset new password			
Email Management			√
- Update email recipient list			
- Send email			
Dashboard		√	
- Recent added issues			
- Recent comment			
- Recent activities			
Web Calendar		√	
- Add event			
- Update event			
- Delete event			
Report		√	
- View by status			
- View by project			
- View by issue type			

4.6 Testing

Testing the system is a very important stage to ensure that all system requirements have been developed without errors. System testing can be done through some stages. The first stage is called unit testing or component testing and it is done during the development of the system. Each component, script or module test isolates from other component or unit by checking the input and output for it. The test strategy consists of a series of different tests that will fully exercise the CBT System. The primary purpose of these tests is to uncover the systems limitations and measure its full capabilities. The list of various

planned tests had been done for the CBT system are Functional Test, User Interface Test, User Acceptance Test and Security Test.

5 DISCUSSION AND RECOMMENDATION

During the system development phase, problems have been encountered and proper solutions have been applied on it in order to solve these problems. Those solutions can be found from various sources, such as books and articles, discussion forum and technical resources web sites. Besides, there are several recommendations for the future researchers in preceding this study in depth. The following recommendations are as follows:

- Multiple languages – Currently there is only one language available, which is English. The system can be enhanced to have multiple languages, such as English, Bahasa Malaysia and Chinese and allow user to choose the preferred language in the system.
- Online help document – The system can be enhanced to have an online help document in order to guide the user to use the system. The online help document includes all the features available, step-by-step tutorial and Q&A section.
- Live chat room – The system can provide another chat room function to allow
- Internal users to communicate with each other instantly. The chat function is embedded in the system. As a result, the chat room is restricted and can only be accessed by registered users.

6 ACKNOWLEDGEMENT

We would like to thank Universiti Teknologi MARA for giving a full cooperation towards the completion of this paper. All the guidelines and resources are very beneficial to the successful of this project. Apart from that, very special thanks to employers in Anjung Penchala Sdn. Bhd. for their kind helps in giving the information for this project during the interview session.

7 REFERENCES

- [1] Abeysinghe, S. 2009. PHP Team Development: Ways of Collaboration. Retrieved January 16, 2010 from <http://library.books24x7.com>
- [2] Page, A., Johnston, K., & Rollison, B. 2009. How We Test Software at Microsoft: Managing Bugs and Test Cases. Retrieved January 15, 2010 from <http://library.books24x7.com>
- [3] Black, R. 1999. Managing the Testing Process: The tools you need. Retrieved January 20, 2010 from <http://library.books24x7.com>
- [4] Doar, M. 2005. Practical development environments. Retrieved February 02, 2010 from <http://books.google.com.my/>
- [5] Limaye. 2009. Software Testing. Retrieved February 06, 2010 from <http://books.google.com.my>
- [6] Kogent Solution Inc. 2007). J2Ee 1.4 Projects. Retrieved February 07, 2010 from <http://books.google.com.my>
- [7] Goldin, L. & Rochell, L. 2002. Software Development Bug Tracking: “Tool Isn’t User Friendly” or “User Isn’t Process Friendly”. Retrieved February 11, 2010 from <http://www.springerlink.com.newdc.oum.edu.my/content/4he0d2ehj6m0y5kv/?p=111918f3b5434b9c979278dae45d93e0&pi=2>
- [8] Henderson, C. 2006. Building scalable web sites. Retrieved from <http://books.google.com.my>.
- [9] Robbins, J. 2000. Debugging Applications: Getting started debugging. Retrieved January 30, 2010 from <http://library.books24x7.com>.
- [10] Singh, L., Drucker, L. & Khan, N. 2004. Advanced Verification Techniques: A SystemC Based Approach for Successful Tapeout. Retrieved February 05, 2010 from <http://books.google.com.my>.
- [11] Smart, J.F. 2007. Javaworld.com: What issue tracking system is best for you? Retrieved February 07, 2010 from <http://www.javaworld.com/javaworld/jw-03-2007/jw-03-bugs.html?page=1>
- [12] Barnson, M.P. 2001. The Bugzilla Guide. Retrieved February 10, 2010 from <http://db.glug-bom.org/Documentation/Bugzilla-Guide/>
- [13] Craig, R. D. & Jaskiel, S. P. 2002. Systematic software testing. Retrieved February 03, 2010 from <http://books.google.com.my>.
- [14] Sweeney, 2006. 10 ways to promote your website. Retrieved February 13, 2010 from [http://books.google.com.my/books?id=80KgKvEAYSUC&pg=PA314&lpg=PA314&dq=Sweeney+\(2006\),+RSS&source=bl&ots=yL24QZ1atf&sig=QVCQzHR46jDIFaIVsVZgI_y8NbM&hl=en&ei=pvm7TKyiFpKGvAOxhNXrDQ&sa=X&oi=book_result&ct=result&resnum=3&ved=0CBsQ6AEwAg#v=onepage&q&f=false](http://books.google.com.my/books?id=80KgKvEAYSUC&pg=PA314&lpg=PA314&dq=Sweeney+(2006),+RSS&source=bl&ots=yL24QZ1atf&sig=QVCQzHR46jDIFaIVsVZgI_y8NbM&hl=en&ei=pvm7TKyiFpKGvAOxhNXrDQ&sa=X&oi=book_result&ct=result&resnum=3&ved=0CBsQ6AEwAg#v=onepage&q&f=false).
- [15] Bugzilla Bug Life Cycle. 2007. Bug Life Cycle. Retrieved June 15, 2010 from <http://www.softwaretestinghelp.com/bug-life-cycle/>
- [16] Telles, M and Meloni, J.C. 2008. PHP 6 Fast & Easy Web Development. Retrieved March 01, 2010 from http://library.books24x7.com/book/id_29953/viewer.asp?bookid=29953&chunkid=0471603613
- [17] Swedberg, K and Chaffer, J (2009). Learning jQuery 1.3: Better Interaction Design and Web Development with Simple JavaScript Techniques. Retrieved February 15, 2010 from <http://library.books24x7.com>.