Analytics-Driven Customer Complaint Management System

A dissertation submitted for the Degree of Master of Information Technology

Name: C.D Munasinghe
University of Colombo School of Computing

2019

Abstract

Defects or bugs have existed as a problem in 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. A huge number 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.it becomes very important to have appropriate bug tracking tool. Bug tracking tool makes communication between teams more effective and all the bugs and changes are recorded in the web-based 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 sections to report issues and understand the latest progress in an issue. It consolidated and organized information in a single system and centralized data repository for the changes and bugs in the system. Besides, this system could be accessed by team members concurrently to improve efficiency and they can search particular information in the system easily.

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: C.D Munasinghe

Registration Number: 2016/MIT/046

Index Number: 16550469

____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr. C.D Munasinghe

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. G.D.S.P.Wimalaratne

____

Signature: Date:

Acknowledgments

Sincere gratitude is hereby extended to Dr. G.D.S.P.Wimalaratne, department of communication and media technologies of the Computer Science Department, University of Colombo School of Computing, for his time and to be a member of my supervisory committee. Also, for his continuous guidance and the compassion he had shown throughout my Master degree program that helped me immensely to achieve my goals.

A special thanks to the faculty of the Computer Science department for all their help and the support at all the time throughout my program.

Table of Contents

Table of Figures

List of Tables

Chapter 1

Introduction

In any organization, customer satisfaction is its main goal. Without the people who buy the product or service, the company will join another group of bankrupt companies. Customers expect immediate solutions without problems from the organization. In a service company, the customer may have complaints, requests, etc. about software products. The organization must ensure that it meets the expectations of its clients as quickly as possible. The goal of this project is to monitor, track, and review all customer expectations in a centralized system using a tool to identify problems and improve business.

Today, organizations face many challenges in solving customer complaints.

Most organizations do not maintain an adequate complaints management system. Where it is done manually currently, in most organizations, a customer who wants to make a request or file a complaint about a particular area will inform the relevant party. At this point, if that person is busy, Particular person can receive the client's request verbally or write it somewhere in their notebook. When time passes, they can't remember half the customer's request, or perhaps completely forget the customer's complaints. They will remember the request once again when the customer calls them again and they will be asked whether they have fixed the problem. This led to customer dissatisfaction.

In addition, customer expectations can be categorized in several stages, some of which have high priority, low priority demand, and so on.

If the priority is high, these issues must be resolved and resolved within one business day. But in the current system, the task can easily be lost because there is no adequate mechanism to track all customer requests. Another problem is that when the complaint is being completed by the development team, the customer does not know the stage of processing the actual application (at the development stage, the quality control stage, etc.) of the request. specific complaint request. In this case, if the person wants to know the current status of the request, they must call or send an e-mail to the corresponding person to check the status of the request. It becomes a job that takes a lot of time.

To overcome these organizational problems, it is necessary to develop a transparent complaints management system, based on an analysis.
The main requirement of the system is that it facilitates the addition of requests (claims, requests), monitors the status of all requests, analyzes customer requests, and also monitors the performance of the staff member using the system. The purpose of transparent complaint management is if a customer files a complaint, a request for a particular area, this system must provide a transparent way to check the status of that request and who is responding to the particular request, etc.
Error tracking is a process of tracking software errors reported in the products. It allows developers to adjust software design by making continuous updates of products or updates to better serve customers. The error and change tracking system is a web-based application that tracks the change request and any errors detected in a system. Once it is added, each item is tracked until it is completed with useful reports to show progress. People have to follow up all the problems because they tend to forget and forget the problems in the system.

The target users of the system are clients, software evaluators, project team members, executives and software developers. It allows multiple users to track and manage multiple projects simultaneously. This system also supports the development and support teams. When the developers complete a task, they can continue working on the following problems while the Quality Control (QA) team begins to perform the tests. The quality control team can use the error tracking system to track the evolution of problems until a developer saves time to review other problems. If the problem is critical, developers can fix it immediately. However, if the problems are not critical, developers can choose to solve them based on the priorities of the activities they are working on now. As a result, it's not just about tracking problems, it also helps the quality control team interact more effectively with developers. Each reported error must be tested and replicated by the quality control team. They must simulate the scenario and produce the results of the test. Maintaining a good record of the decisions made will help to control the scope of the project. Error tracking also lets user know what has been done with the problems identified, why the system was changed, and what changes should be taken into account. The system is compatible with multiple users and multiple users can access information simultaneously from any type of computer without installing software on the user's computer. As a web application, users can collaborate from anywhere, as long as they have access to the Internet and a web browser. This project proposed an update of the error and change tracking system for any service-based organization.

1.1 Aims and Objectives

This system is focus to implement a single platform to cater to ERP functionality and manage bugs/change requests. In return, this will provide a smoother application to the user. The system will help top management to overlook ongoing tasks and make decisions some of the key considerations are,

- What are currently ongoing tasks?
- What is the current progress of each task?
- Who is responsible for a particular task?
- What are the task priorities?

By the project, our objectives are to achieve the following goals

1. Provide the ability to track defects/problems
2. Provide project-level data entry support
3. keep track of employee skills and based on the skills assigning of the task.
4. Generate various report by the system for an employee and as well as to a manager to track bugs/issues.

1.2 Scope

A bug tracking system is a perfect or unique solution for tracking application errors. The error tracking system allows one or more developers to continue tracking unfinished errors Or any request for a change in their software products. For good user experience, the error tracking system can help a lot, increase the accountability and productivity of each employee by giving positive feedback and supporting the workflow. The error tracking software allows a group of individual testers or testers to keep track of unresolved errors in their software. The error tracking software can track errors and monitor the current status of each error in its life cycle.

Chapter 2

Background

The main goal of this project is to provide a convenient user-friendly bug tracking application, a simplified error tracking system. Users will not feel uncomfortable using this system to track problems, minimize steps and provide clean navigation throughout the system. The software development project faces multiple challenges to meet all the requirements of customers and users. Meanwhile, the software project must identify the errors or defects, correct them, in the current system. Defects or errors are problems in the system and are usually unavoidable in software development. A software error is a term commonly used to describe an error, defect, failure in a program or computer system that produces an incorrect or unexpected result or causes unexpected behavior. Most errors come from errors and mistakes made by people, either in the source code of a program or in its technical design.

The purpose of tracking complaints/errors is to improve the quality of software products. Errors can cause the system to perform a task incorrectly or cause events that should not occur. This can be particularly problematic for users who cannot reach certain areas of the system or perform certain operations on the system due to an error in the system. Once the software is distributed to customers, all organizations must adhere to a customer problem management process with the ultimate goal of customer satisfaction. It is more difficult to start tracking errors during the software development phases since software developers do not show much cooperation to find their errors. They believe that it threatens both their creativity and their professional ego. Previously, people followed the error in a simple way. Some users have reported errors when sending an email to a technician or related business analyst. Some users have reported errors over the phone. Sometimes, some complaints are made on the spot. It is quite difficult to track the reported error because there are many ways. The easiest and most efficient way to track a small number of errors is to use a spreadsheet program, such as Excel. The main advantage of this approach is its simplicity and it is suitable for a small project.

The drawbacks of this approach are that, as the project develops, the first problem that can be found is that only one person can modify the spreadsheet at a time, users can not modify the spreadsheet simultaneously and cannot effectively use the spreadsheet application. Another scenario is that the total number of errors in the spreadsheet reaches thousands, the time needed to find and manage the spreadsheet is more difficult. For this reason, an error tracking system is created to help the quality control team and the development team track error reports. There are many simple and inexpensive error tracking systems available that are better than a spreadsheet.

When a problem is reported, it can involve many tasks such as searching, gathering information, testing and debugging throughout the process. It is quite uncomfortable to handle the problems of a project to track each result in the documents because hundreds of errors can be found. He notes that the developers create errors because the system developers will need more efficient communication.

There are many similar error tracking systems available for this purpose; However, some of them are created with complex and confusing useless functions. The flow is not simple is the main concern. This research study has three main objectives that were achieved at the end of the survey.

The proposed system should provide users with a platform to report issues and understand the latest state of their problems in a single system and a centralized data repository. Without an adequate system, project stakeholders cannot easily access this information.

2.1 Review of similar systems

When a problem is reported, it can involve many tasks such as searching, gathering information, testing and debugging throughout the process. It is quite uncomfortable to handle the problems of a project to track each result in the documents because hundreds of errors can be found. To achieve good results the system developers will need more efficient and effective communication.

There are many similar error tracking systems available for this purpose; However, some of them are created with complex and confusing useless functions. The flow is not simple is the main concern. This research study has three main objectives that were achieved at the end of the survey.

The proposed system should provide users with a platform to report issues and understand the latest state of their problems in a single system and a centralized data repository. Without an adequate system, project stakeholders cannot easily access this information.

2.2 Common features of the existing bug tracking systems

1. Reporting facility – by filling relevant fields a user can provide information about the bug, environment, module, severity, screenshots, etc. [2]
2. Assigning – the related issue can be assigned to users [2]
3. History/work log/comments
4. Reports – graphs or charts

Many companies use a bug tracking system in developing software products because it is extremely useful and valuable. There are numerous bug tracking tools available both commercially and as open source. Selecting the best bug tracking tool that fulfills the users' requirement is not simple. There are many factors that need to be considered as following [2]

- Application Installation

  Bugzilla is probably the most well-known of the open source issue-management tools. It is used in many open source projects such as Eclipse, Mozilla, but sometimes freeware bug tracking tools might be difficult for the user to configure. And not that much user-friendly.

- User-friendliness of the application

  There is so many bug tracking software out there some of them are designed for the purpose of bug tracking. But the problem is some of them are paid ones and also, they are way too complex to manage

- Centralized system

  The system should be a centralized system .user log in to the current system and if the user encounters a bug by the same system should be able to raise a ticket to the issue without switching to any other 3rd party applications.

2.3 Existing Bug tracking software

1. Bugzilla
Bugzilla has been a leading bug tracking tool widely used by many organizations for quite some time now. It is a very simple to use, web-based interface. It has all the features of the essence, convenience, and assurance. It is completely open sourced and is free to use.
Bugzilla is an open-source bug tracking software solution that enables software developers to track bugs and defects during software development. Features include automated bug detection, bug lists, patch management, and web services [3].
Bugzilla's advanced Search feature allows software developers to create advanced search queries to track and list bugs. In addition, the platform shares automated email notifications whenever the bug tracking settings are changed. Bugzilla's Time Tracking feature enables executives to estimate time for debugging, track work hours spent and establish deadlines. It also shares bug list in extensions, such as Atom, Comma-Separated Value (CSV) and iCalendar. Bugzilla allows managers and team leaders to assign and delegate bugs to different team members with the help of a drop-down list [3].
Bugzilla is available in multiple languages such as Spanish, Japanese, French, and German. Bugzilla does not offer much customization options, but the system is powerful and effective in tracking bugs in software and hardware as well as IT support queues and deployment management.
The application delivers an optimized database structure that results in improved performance and scalability. The system features top-grade security to protect the confidentiality of its users. Other features include an advanced query application that can remember users searches and integrated email capabilities [4].

Advantages of Bugzilla

- Open source Product.

  Bugzilla is an open source product, therefore, no need to have a managed license [5].

- Well, manageable defects and reports can be exported.

- Defects can be managed efficiently and Bugzilla has a very advanced reporting system [6].

- Multi-project issue tracking capable

  If an organization is working on multi-project environment Bugzilla can accommodate that kind of situation [6].
- Advanced search capabilities [5].
- Email notifications controlled by user preferences [5].

Cons of Bugzilla

- Complicated user interface

  Though Bugzilla is a well-reputed issue tracking system with lots of functionality user can easily confuse with the user interface. The user interface is not user-friendly [6]
- limited customization options.

  Bugzilla is lacking in filtered reporting areas. [6]

2. Jira [7]

   Jira is a combination of bug tracking, issue tracking, and project management software, Jira provides an advanced reporting system and also it is capable of managing workflows. Jira is not only concentrated on the software development industry but also it works as a help desk, leave management system, supports agile projects, etc. Jira is developed by Atlassian Jira has 60% market share on the bug tracking software category.

   Jira is designed to help users capture, assign, and set priorities to their work. It allows a user to manage the whole process of application development making sure that all things are covered, from concept to launch. It's simple, intuitive interface enables collaboration with teammates and allows to get the job done in an effective manner.

   At the moment, Jira is used by more than 51,000 customers worldwide including top brands like eBay, Spotify, Cisco, and LinkedIn. It is a leading software development app used by agile teams. Team members can use Jira to plan, track, and release effective software. User can create user stories and issues, plan sprints, and assign tasks to team members. Prioritize and discuss with team's tasks in full context with visibility. Plus, Jira offers real-time, visual data reports to boost team performance. [8]

Advantages of Jira [9]

- Can use for any type of industry [9]

  Jira can be used by any type of company.it has so many features that can be used in any kind of organization. It can work as a bug tracking software, work log manager, help desk Jira can be used in many ways according to the organization

- Project task management and time tracking

  Jira is capable of managing workloads, project task. Each task can be monitor individually and also can log the time for each task

- Advance agile project management features [10]

  JIRA has all advanced features an agile project management tool provides like scrum tracking, Sprint burndown chart, Agile reports, velocity charts. JIRA supports estimation of tasks in points as well as hours and does also track the actual time spent on the task.

Disadvantages of Jira [11]

- Complex user interface

  Jira has a complex user interface for a new user it may be a big learning curve to learn the software
- Not an opensource software

  Jira is a paid software. Pricing is based on the user wise and monthly subscription fee have to pay

2.4 Comparison of the existing system

Systems	Bugzilla	JIRA	MantisBT
Description	Bugzilla is an open source web-based bug tracking tool originally developed by the Mozilla project.	Bug tracking and project planning tool, available as a cloud service and as a commercial tool.	MantisBT is an open source issue tracker written in PHP.
Category	Issue Management	Project Planning Issue Management	Issue Management
License	Open Source	Commercial Web-based service (SaaS)	Open Source
Web-based architecture	Yes	Yes	Yes
Features	Workflow Management	Agile planning Scrum and Kanban board task prioritization Real-time release tracking	Issue Tracking Workflow Management

2.5 Expected system

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 system should be a web-based one. All of the above applications are 3rd party tools where the user wants to log in to a separate system in order to make a complaint, track or update activity. Users may hassle about using two systems. The main expectation of this system is company has its own ERP software as a product. There are many clients using this software for their day to day business activities. Along with users works they might be facing a bug or feature requirement to the existing software so they should be able to raise a ticket for the particular request or a bug. User doesn't have to log in to another system and also doesn't have to manage and pay for any 3rd party systems.it made organization easy where they do not want to manage another software tool for bug tracking and it saves organization money.

Chapter 3

Methodology

Errors in the software development process are inevitable. However, they must be tracked and corrected, otherwise, they will be manifested in the final product, which will cause a failure in the software systems. The life cycle of software development consists of several phases. The development team can make mistakes at any stage. However, when the time comes for the code, it appears once the execution is tested. Most errors can be avoided by having a correct analysis and design according to the client's requirements. Depending on the software errors, the system fails.

3.1 Issue Tracking Life Cycle

The system design was proposed based on the needs of the user and the new retail system. Each problem in the system will have an associated status, priority, and severity. The status indicates the current progress against a problem.

Issue Status

Issue Status	Description
Open	There is an issue/bug available to fix
In-Progress	Someone is working on the issue
Resolved	Someone provided a solution to the issue
Closed	Someone verified the solution and the issue is fixed

Table 3.1 Issue Status

When an error is found, a problem is created in the problem tracking system and the problem is in the Open state. Once a particular user has started working on the problem, it will be transferred to Instate. When the developer provides a solution, the problem is that the solution must be verified. If the verification fails, the problem will reopen and return to the

Open state. If the verification is successful, the problem can be closed, indicating that the error is solved (Table 3.1 Issue Status).

The priority indicates the severity of the problem. High priority issues must be addressed before others. The severity indicates the severity of the problem. In other words, gravity indicates the impact of the problem on the system.

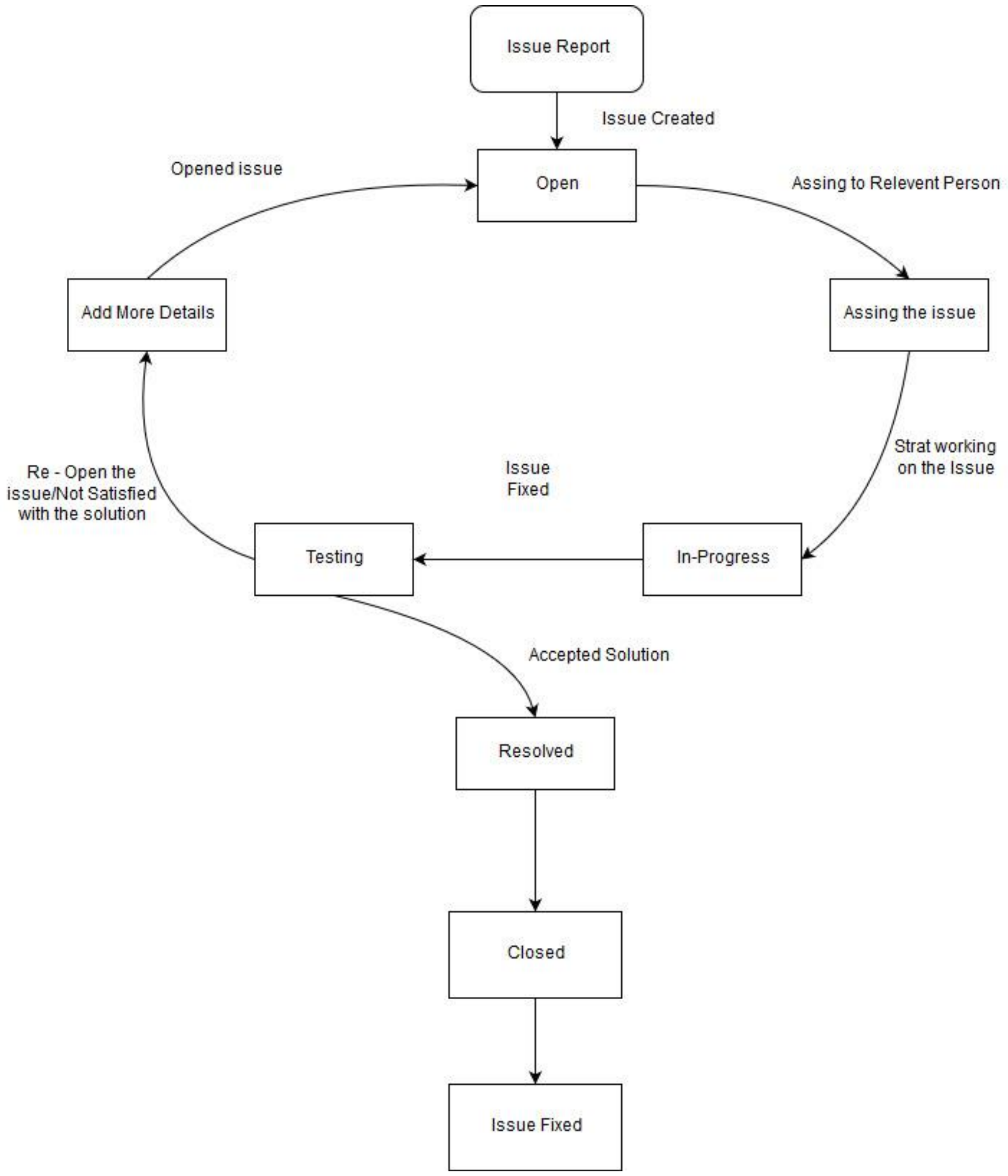


Figure 1.01 Issue Tracking Life Cycle

When a person wishes to file a complaint or report an error, they must log in to the current ERP system and may file a complaint at that time. It can even designate a person to correct the error. Then, once an error is generated, they can be considered as an open state in the error. Then, a concerned person can take this mistake and correct it. At the time the error is reported, the reporter may also mention the severity of the problem reported by the criticality level developer who will handle these issues based on the level of criticality. If it's critical, the priority is high. When someone assigns a particular problem state, it moves from the open state to the assigned state. Then, when the affected person starts working on the error, the problem status changes from the assigned status to the InProgress state.

Then, after correcting the error of the affected person, mark the problem as corrected, and replace the current states with Tests. Then, the responsible quality control manager will be in charge of the problem and will check if the solution provided is acceptable or not.

If the solution provided is not acceptable, return the problem status to open and assign it to the same person. If the solution provided is acceptable, mark the problem as resolved and close it (Figure 3.01).

3.2 Overall flowchart of Bug Tracking System

In the complaint management system, there is a different type of user and user permissions. Normally, (Figure 3.02) all users must first access the login page and use their credentials to log in to the system. If the credentials are correct, they are directed to the main ERP home screen. Then, if user has an error to report, a complaint or any suggestion, any user can simply access the problem management interface and file a complaint. Once the complaint is filed, an automated email is sent to assign a person or administrator the details of a complaint and the number of the complaint. Then, the normal life cycle of problem tracking will continue.

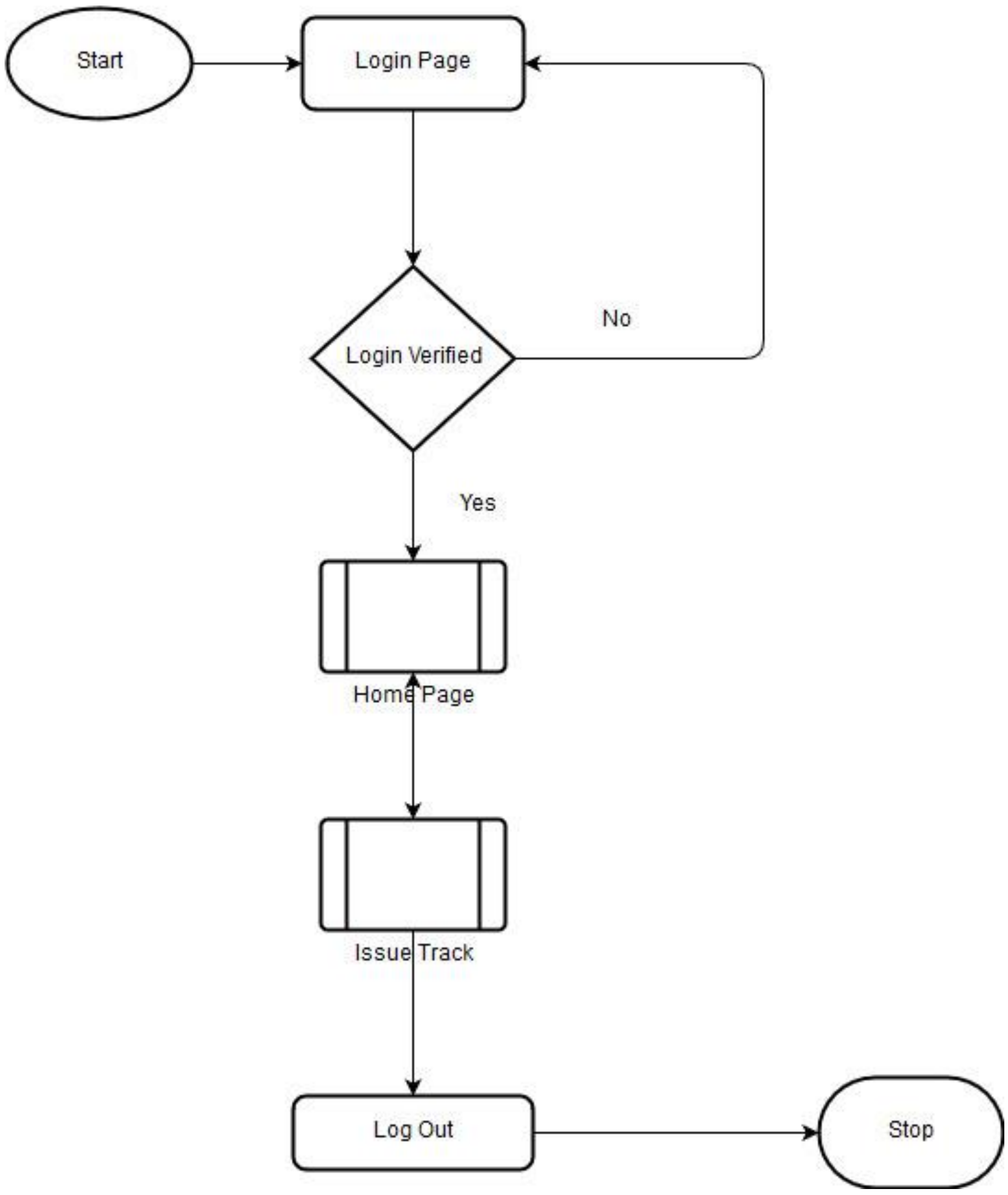


Figure 3.02 flowchart of Bug Tracking System

3.3 User Roles and their responsibilities

3.3.1 Admin Role for Issue management

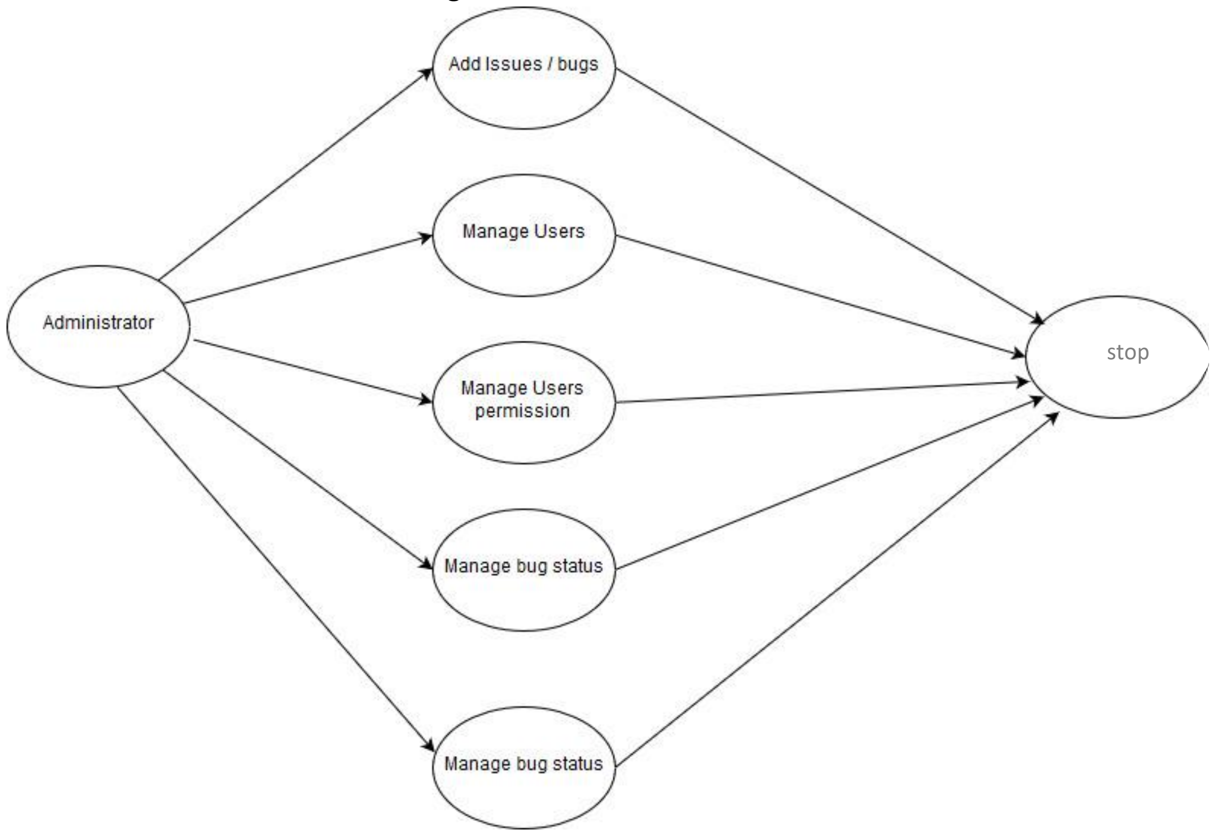


Figure 3.03 Admin Roles and their responsibilities

3.3.2 User Role

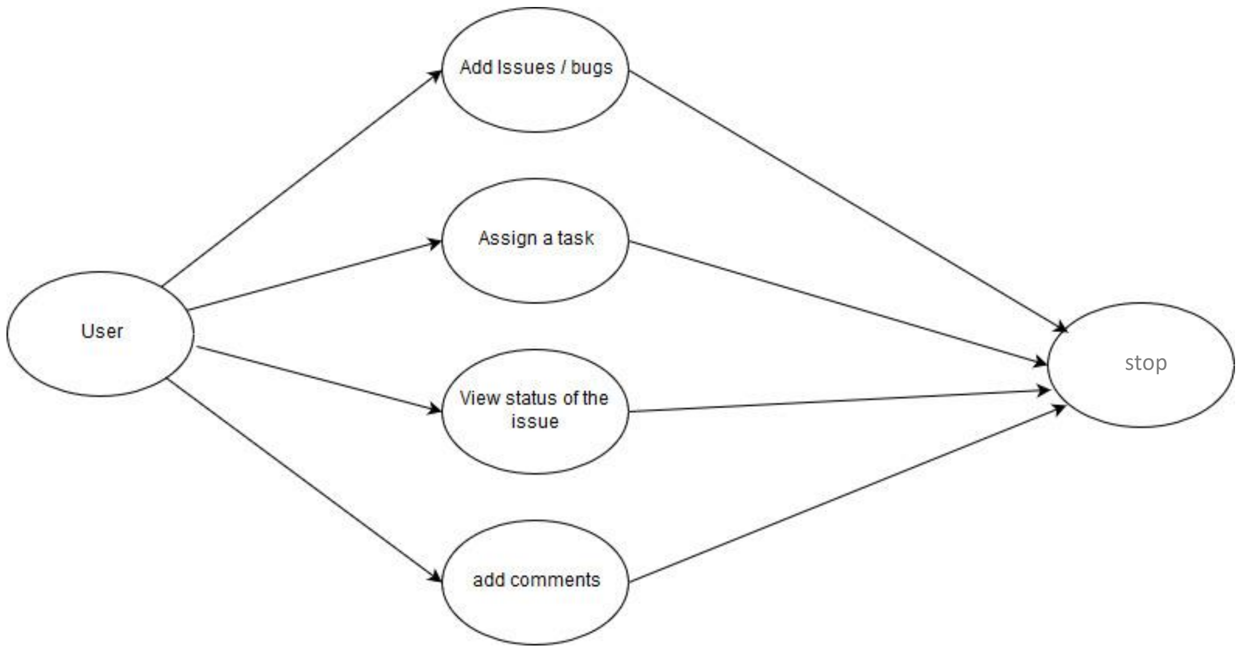


Figure 3.04 User Roles and their responsibilities

3.3.3 Developer Role

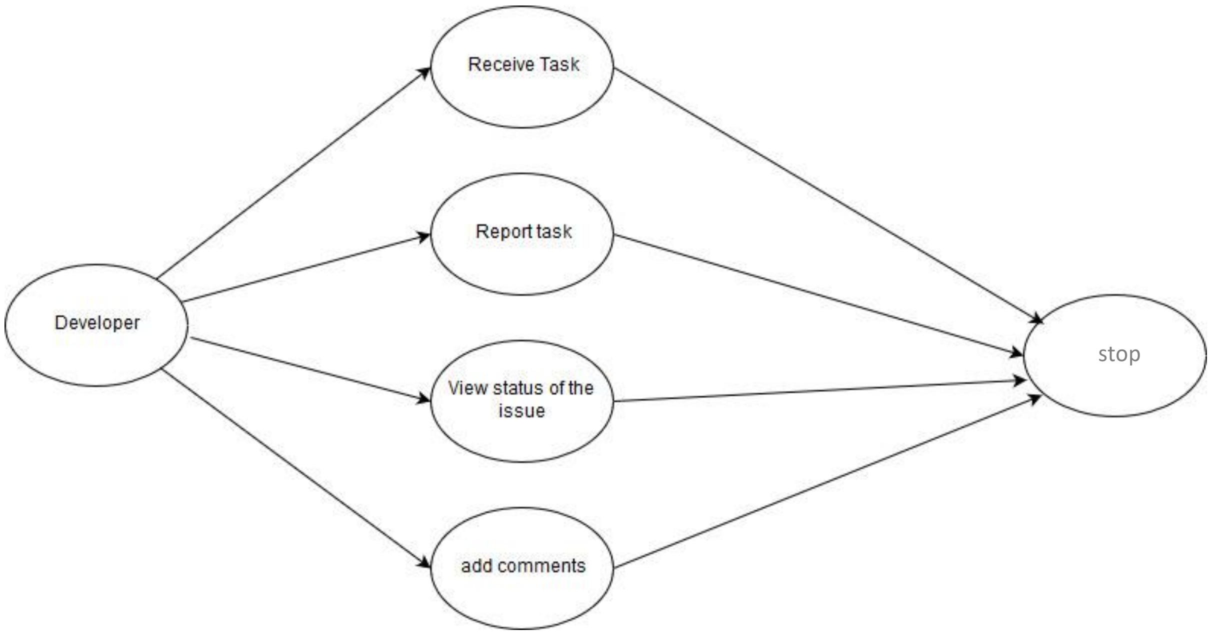


Figure 3.05 Developer Roles and their responsibilities

3.4 Technologies

ASP.Net with MVC Architectural pattern. Kendo UI, JavaScript, jQuery used as a front end client-side scripting languages and the backend is a SQL Server is used.

ASP.NET

ASP.NET is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications, and web services.

Kendo UI

A JavaScript framework, a collection of JavaScript UI components with libraries for jQuery. Build eye-catching and high-performance responsive web applications choice.

SQL Server 2017

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network

ASP.NET Web API

The ASP.NET Web API is an infrastructure that facilitates the creation of HTTP services that reach a wide range of clients, including browsers and mobile devices. The ASP.NET Web API is an ideal platform for creating RESTful applications in the .NET Framework. All users will connect to the database via user interfaces. These interfaces are based on four main modules.

Issue management - This module creates and manages the problem in the system. Several problems can be created and users can report a problem. Every problem reported in the system has its status to indicate progress. The user can handle the details of the problem, such as status, comments, history, attachments, and comments. The system maintains a history of problems that indicate that changes have been made to a problem during its lifetime and users can see the history associated with the problem.

User management - This module is used to create and manage users in the system. The administrator can create an unlimited number of members with user profiles. store encrypted passwords in the database and limit certain components to certain user groups. Users have created recategorization as customers, software evaluators, project team members, executives, and software developers. Each category of users has different access rights in the system because they assume different roles and responsibilities in the progression of software development.

Email management - Users with the specified access right can determine the recipients of the email and assign the selected email addresses to the products accordingly. An automatic email notification will be sent to the associated users based on the list of recipients of the email when the problem is reported.

Dashboard – A general description is provided so that users know what are the outstanding issues and what is the status of the final description of the graph-based user complaint.

3.5 Function Architecture

The functionalities of complaint management are categorized in the following modules such as Issue management, user management, email management, and dashboard (Table 3.02).

Function	Member	Admin
Issue Management		√
Add an Issue	√	
Update an Issue	√	
User Management		
Register new user		√
Update user details	√	
Change password	√	
Reset Password		√
Email Management		
Update email recipient		√
Send email		√
Dashboard		
Complaint status Dashboard	√	√

Table 3.02 Function Architecture

3.6 Software architecture

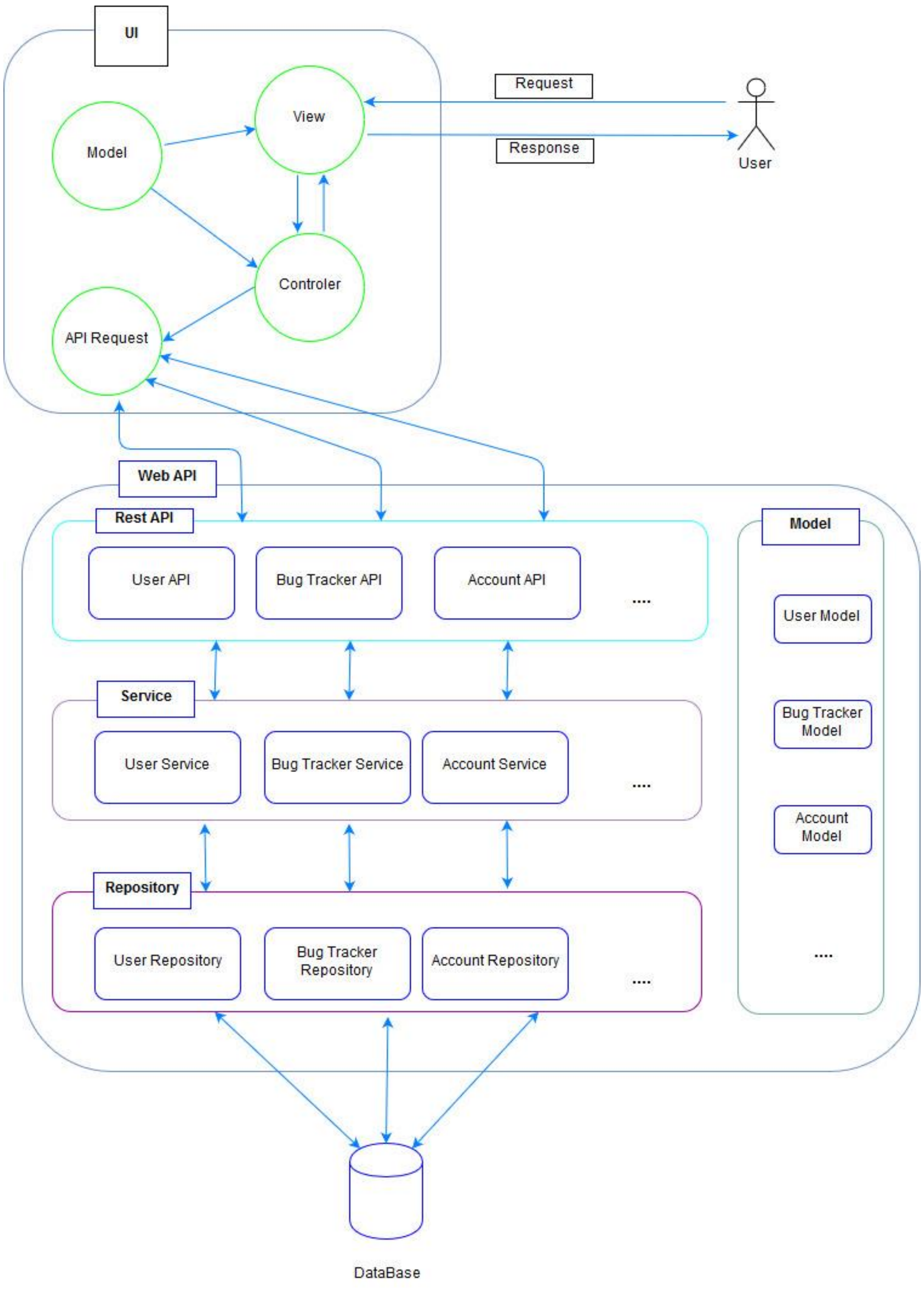


Figure 3.06 Software architecture

Software is consisting of two layers (Figure 3.06). UI is one part with is having MVC architecture and the second layer is WebAPI layer.

UI Layer

This layered work as a front-end framework for the application. MVC framework is used to develop the application. ASP.net razor technology is using in view section to render data from server-side code. Other than razor syntax in this layer JavaScript, Ajax, Kentico UI API, bootstrap was used to develop the view model. When a request came from the user using MVC framework functionality request is modified and generate API request in the API Request module and sent a request to Web API.

WebAPI

WebAPI Layer is developed using REST API, ASP.NET Web API 2 is used as the WebApi technology. This layer is divided into multiple sections. They are categorized as Web API controller layer, Service layer, Repository layer

Web API Controller

When API request is received to the WebApi controller by the controller layer itself validates the API request using an API request token. If it is a valid request, Request is pass to the service layer

Service Layer

A Service Layer defines an application's boundary and its set of available operations from the perspective of interfacing client layers. It encapsulates the application's business logic, controlling transactions and coordinating responses in the implementation of its operations

Repository Layer

The Repository pattern is used to decouple the business logic and the data access layers in the application. The Repository access layer typically contains storage-specific code and methods to operate data from the data storage server. Stored Procedures are used to execute SQL command.

Database

SQL server use as a database engine

3.7 Database ER Diagram

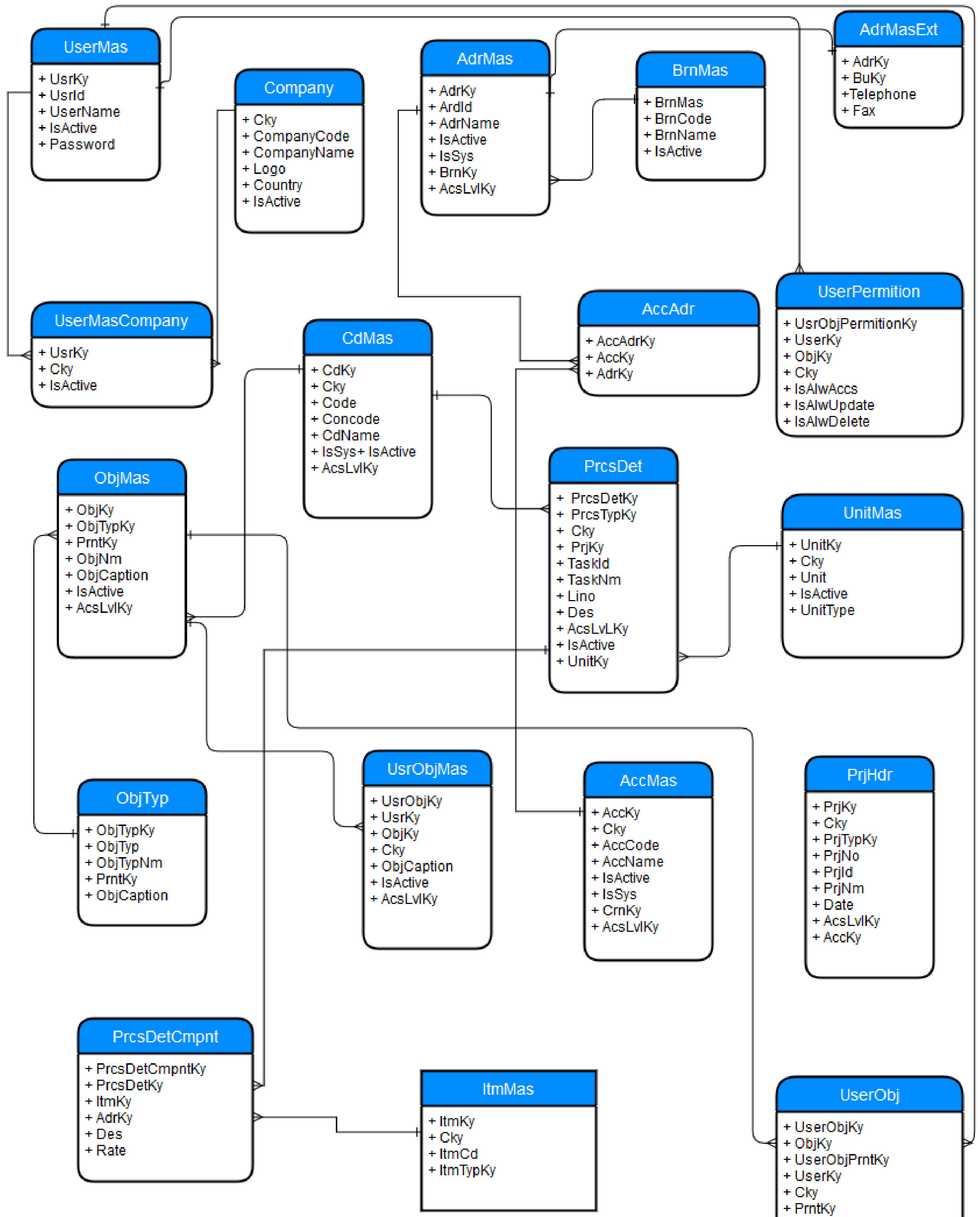
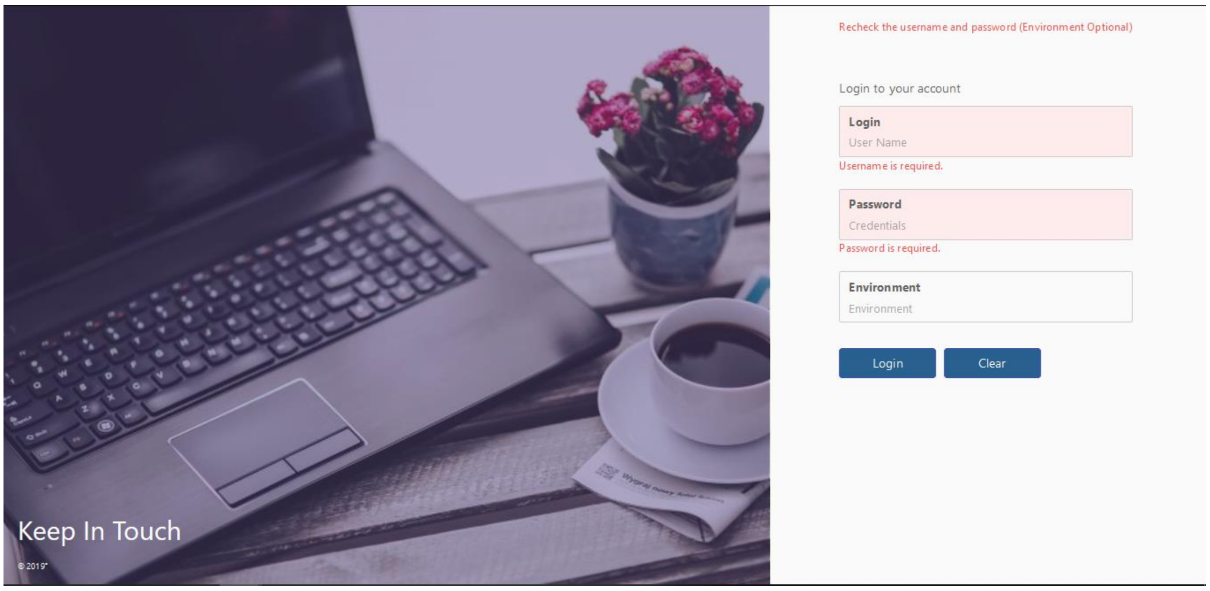


Figure 3.07 ER Diagram

3.8 User Interfaces

3.8.1 Login Screen



Recheck the username and password (Environment Optional)

Login to your account

Login
User Name
Username is required.

Password
Credentials
Password is required.

Environment
Environment

Figure 3.08 login screen

When User Login to the system user have to enter username and password. This application is supported in Multi-Tenant environment. a single instance of the software runs on a server and serves multiple tenants. In that case, the only user has to enter the environment name and login to the system. Upon successful login, the user will be direct to Company selection menu if the user has registered for more than one company. Or else if a user is registered to one company directly user will be direct to the user menu (Figure 3.08).

3.8.2 Company Selection Screen

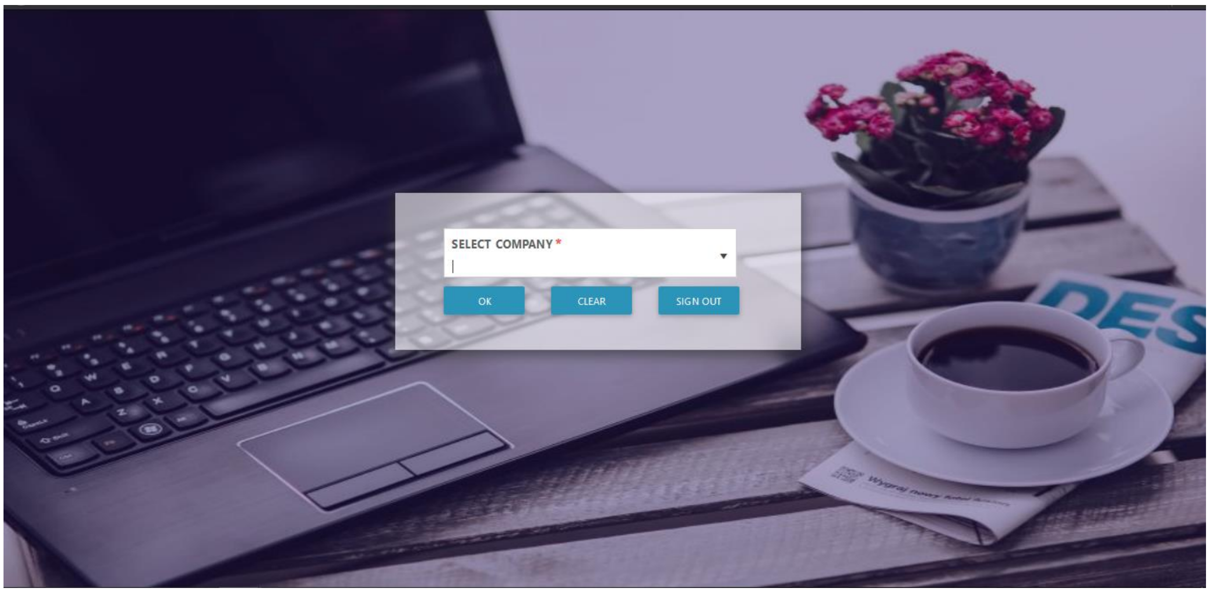


Figure 3.09 Company Selection Screen

Upon successful registration, if the user is registered for more than one company user will direct to this screen (Figure 3.09) to select a company. Based on selected company users' values and menu item might be changing based on user permissions.

3.8.3 Menu Selection Screen

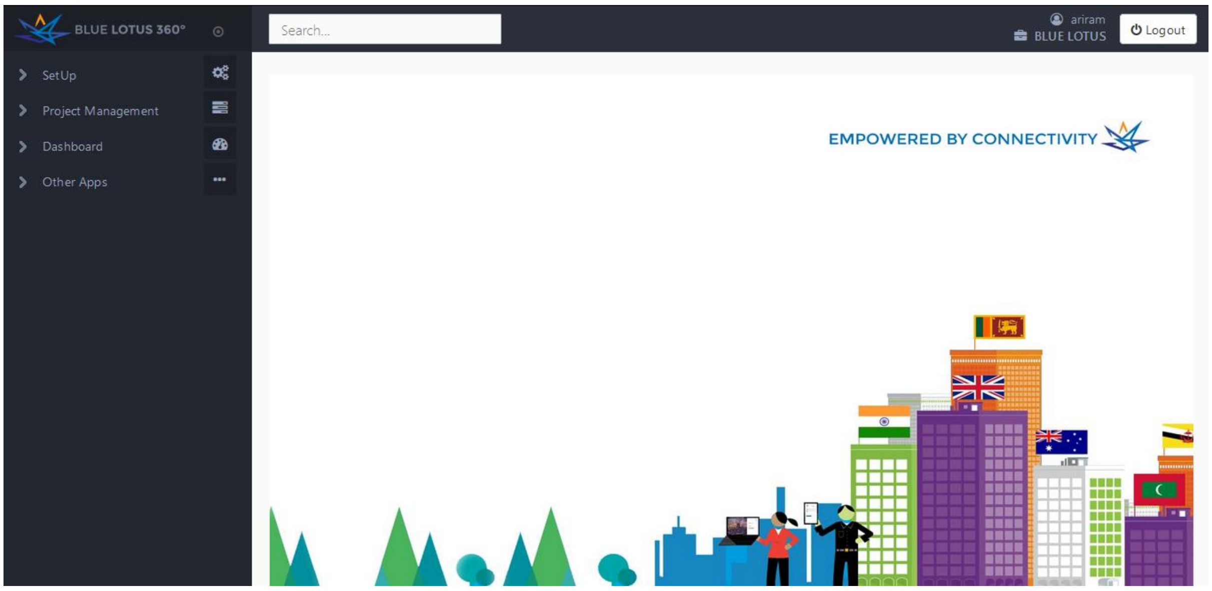
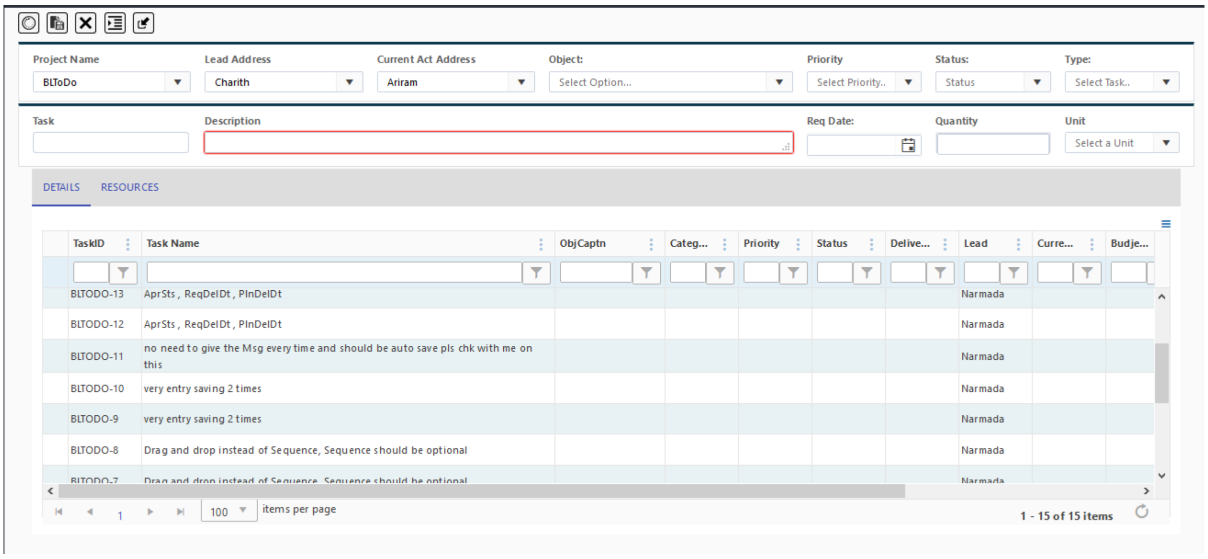


Figure 3.10 Menu Selection Screen

Based on user Permission menu will be shown to the user (Figure 3.10). If the user has admin permissions user can configure other user permissions.

3.8.4 Customer Complaint Management Interface



Project Name	Lead Address	Current Act Address	Object:	Priority	Status:	Type:
BLToDo	Charith	Ariram	Select Option...	Select Priority..	Status	Select Task..

Task	Description	Req Date:	Quantity	Unit
				Select a Unit

DETAILS RESOURCES

TaskID	Task Name	ObjCaptn	Categ...	Priority	Status	Delive...	Lead	Curre...	Budje...
BLTODO-13	AprSts , ReqDelDt , PlnDelDt						Narmada		
BLTODO-12	AprSts , ReqDelDt , PlnDelDt						Narmada		
BLTODO-11	no need to give the Msg every time and should be auto save pls chk with me on this						Narmada		
BLTODO-10	very entry saving 2 times						Narmada		
BLTODO-9	very entry saving 2 times						Narmada		
BLTODO-8	Drag and drop instead of Sequence, Sequence should be optional						Narmada		
BLTODO-7	Drag and drop instead of Sequence, Sequence should be optional						Narmada		

1 100 items per page 1 - 15 of 15 items

Figure 3.11 Customer Complaint t Management Interface

This interface (Figure 3.11) is used to add, update complaint s. According to the selected field, the above grid will be changed accordingly. All the dropdowns are populated from the database.

3.8.5 Customer Complaint management dashboard.

From the dashboard, it gives an overall idea about individual project wise, person wise filtered result into a pie chart (Figure 3.12, Figure 3.13).

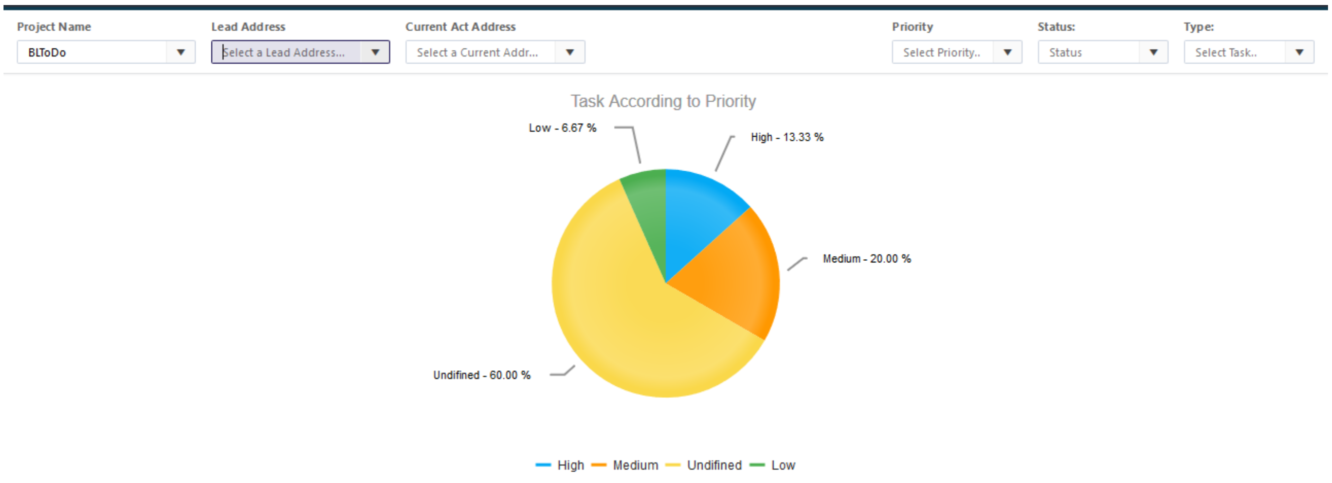


Figure 3.12 Customer Complaint Management dashboard.

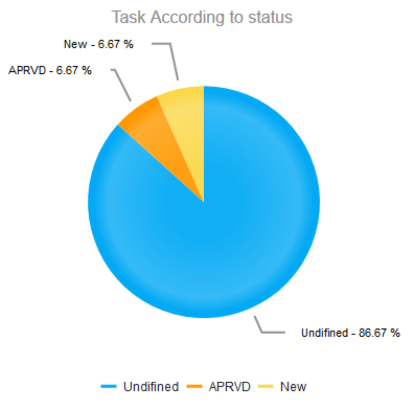
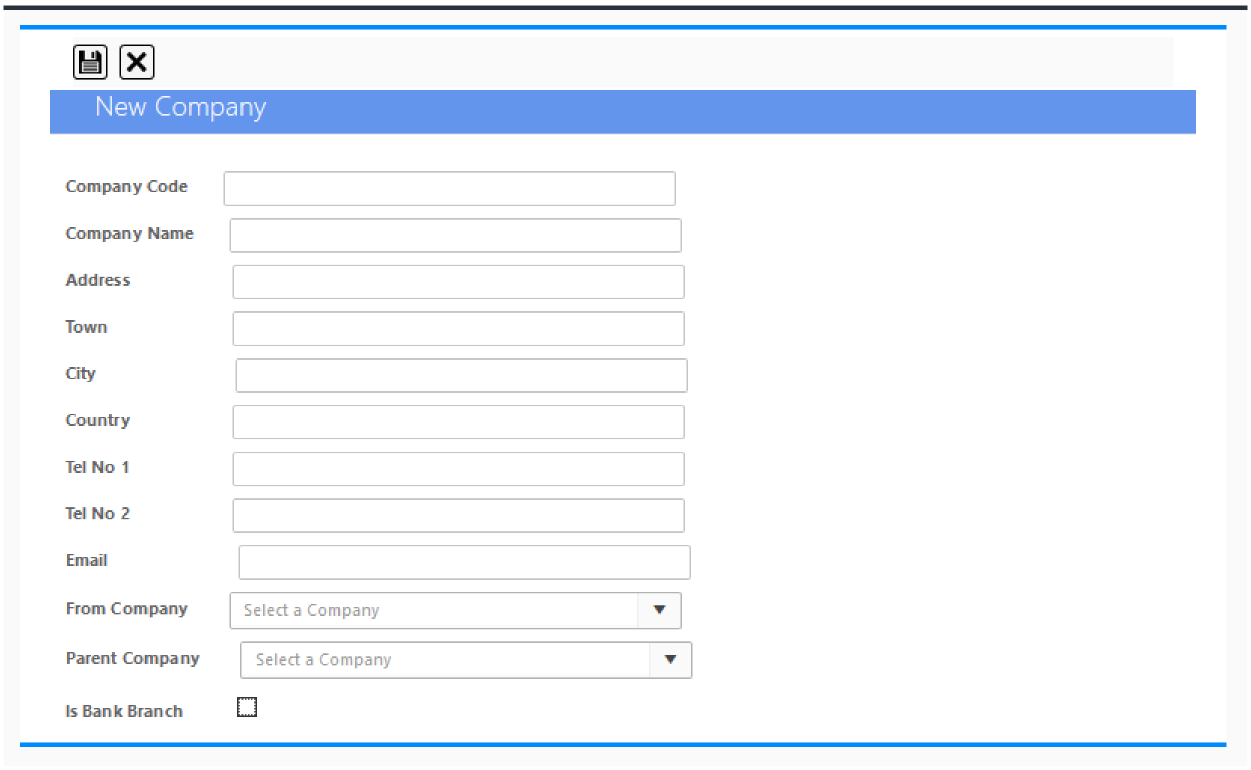


Figure 3.13 Customer Complaint Management dashboard.

3.8.6 Main data entry user interfaces

3.8.6.1 Register a new company to system.

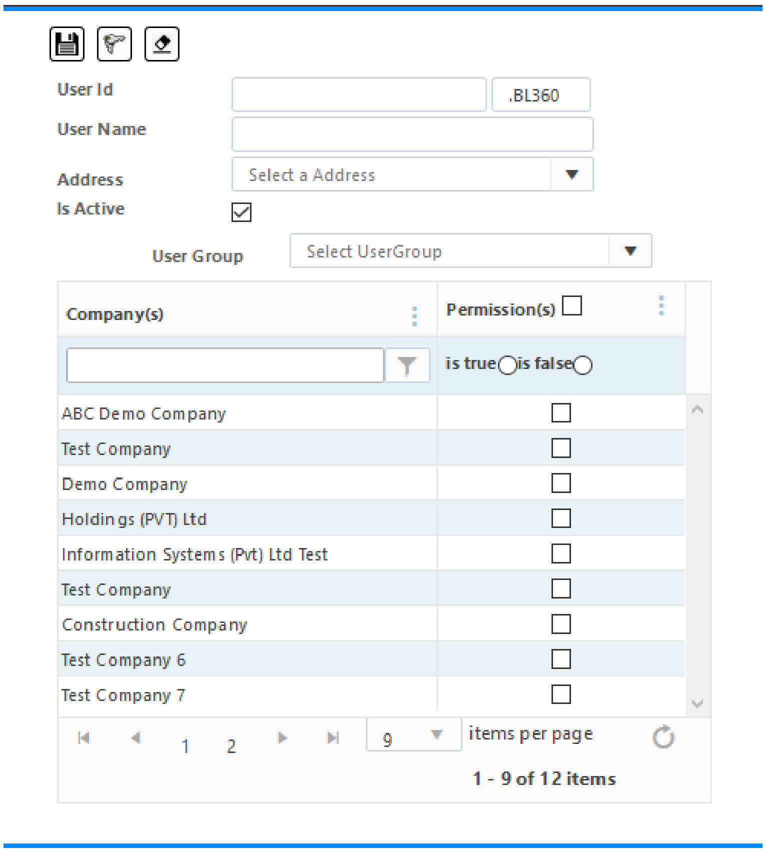

New Company

Company Code	
Company Name	
Address	
Town	
City	
Country	
Tel No 1	
Tel No 2	
Email	
From Company	Select a Company
Parent Company	Select a Company
Is Bank Branch	☐

Figure 3.14 Register a new company to system

This user interface (Figure 3.14) is used to create a new company. From company and parent company is used to copy from another company. It works as a template to create a new company. When a new company is created automatically all the necessary related components are being created from the database.

3.8.6.2 New User Register



User Id	.BL360
User Name	
Address	Select a Address
Is Active	☑
User Group	Select UserGroup

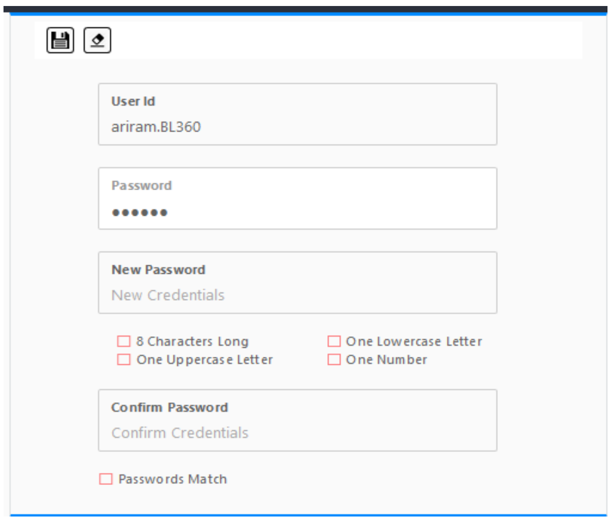
Company(s)	Permission(s) ☐
	is true ◯ is false ◯
ABC Demo Company	☐
Test Company	☐
Demo Company	☐
Holdings (PVT) Ltd	☐
Information Systems (Pvt) Ltd Test	☐
Test Company	☐
Construction Company	☐
Test Company 6	☐
Test Company 7	☐

1 2 | 9 items per page

1 - 9 of 12 items

Figure 3.15 New User Register

3.8.6.3 User Password Reset



User Id
ariram.BL360

Password
••••••

New Password
New Credentials

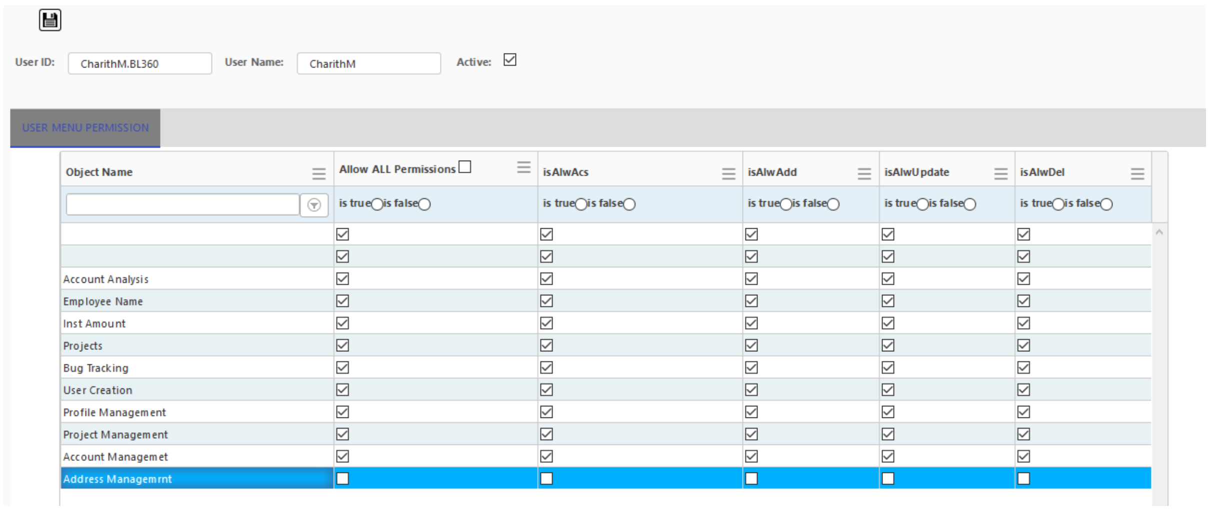
☐ 8 Characters Long ☐ One Lowercase Letter
☐ One Uppercase Letter ☐ One Number

Confirm Password
Confirm Credentials

☐ Passwords Match

Figure 3.16 User Password Reset

3.8.6.4 Set user permission



User ID: CharithM.BL360 User Name: CharithM Active: ☑

USER MENU PERMISSION

Object Name	Allow ALL Permissions ☐	isAlwAcs	isAlwAdd	isAlwUpdate	isAlwDel
	is true ○ is false ○	is true ○ is false ○	is true ○ is false ○	is true ○ is false ○	is true ○ is false ○
	☑	☑	☑	☑	☑
	☑	☑	☑	☑	☑
Account Analysis	☑	☑	☑	☑	☑
Employee Name	☑	☑	☑	☑	☑
Inst Amount	☑	☑	☑	☑	☑
Projects	☑	☑	☑	☑	☑
Bug Tracking	☑	☑	☑	☑	☑
User Creation	☑	☑	☑	☑	☑
Profile Management	☑	☑	☑	☑	☑
Project Management	☑	☑	☑	☑	☑
Account Managemet	☑	☑	☑	☑	☑
Address Managemrnt	☐	☐	☐	☐	☐

Figure 3.17 Set user permission

User permission can be set using above interface, user read, write update permission can be set by this form (Figure 3.17).

Chapter 4

Evaluation

The test is one of the main steps in the development process in which the project is monitored for the different sets of data and to determine the cases in which the project failed. Therefore, it is useful to choose the steps necessary for the system to be immune to this type of data entry in the future. The test is essential to the success of any software. The evaluation of the software is essential. The tests are also carried out in several phases. The first phase is during software designing, that is, when the module is created. Then the other test phases are performed. Once the software is finished, other test phases are carried out. This is a system test that verifies that all programs work correctly as needed. Some of the tests performed to evaluate the product are mentioned below.

4.1 Black Box Testing

The details of the implementation and the knowledge of the internal routes of the software are now managed by the users. Test is entirely based on the requirements and specifications of the software. In this phase only check for software, functions. The program evaluates only as of the end-user, does not perform internal code analysis. For this scenario, an existing QA organization used for the evaluation.

Using the black box testing attempt to find errors in the following categories

- Incorrect or missing functionality
- User interface Errors
- Error in data structures
- Behavior or performance errors.
- Initialization and termination errors

Following testing, techniques are used for testing

- Equivalence Partitioning
    - involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data
- Boundary Value Analysis:
    - This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases.
- Decision Table Testing:
    - A decision table puts causes and their effects in a matrix. There is a unique combination in each column. According to the decision table values program is evaluated

4.2 White Box Testing

White Box testing was carried out with organization QA team and peer developers. White box testing was carried out because of to identify followings

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- The functionality of conditional loops
- Testing of each statement, object, and function on an individual basis

The white box test is done throughout the implementation of the software, as well as unit test levels. For this product, Company decided not to write an automated unit test case. The only test level of the white box was followed.

The peer developers of the company examine the implementation code of a given field in an error tracking system, determine all the entries and the illegal entries (valid and invalid) and verify the results comparing them with the expected results. At this point, peer developers tend to analyze the implementation code. In addition, the code is reviewed by highly qualified developers who are knowledgeable about secure coding practices, which leads people to evaluate the source code of the product in question. The tester must be able to detect security problems and prevent attacks from hackers and naive users who can inject malicious code knowingly or unintentionally into the application. This was the main reason to have a white box for the product.

The main objective of the white box test is that it is possible to prove that the code can be performed in the early stages and that the test is more exhaustive, with the possibility of covering as much as possible.

4.3 Unit testing

Product each module is considered independently.it focuses on each unit of software as implemented in the source code. This is done as a white box testing. For this project, there is no automated unit test written. All the unit test is done manually.

Following scenarios were considered in unit testing

- The functionality of the entire module/Forms
- Validation for user input
- Checking the coding standards
- Testing the module with all the possible test data
- Testing of the functionality involving all type of calculation
- Commenting on standards in the source code

After completing the unit testing of all the modules, the whole system is integrated with all its dependencies in those modules. Developer team integrated the modules one by one and test the system at each step. This helped n reduction of errors at the time of the system testing

test cases that are used to test the product are below in the “appendix section”.

4.4 Validation testing

The main reason for developing this system is to maximize product results and meet the specified commercial requirements. To obtain a more accurate image, validation tests were performed to meet the specified commercial requirements.

Validation and testing ensure that the product really meets the needs of the customer. It can also be defined to demonstrate that the product meets its intended use when implemented in a suitable environment. By answers, the following question: are we building the right product? (Figure 4.01)

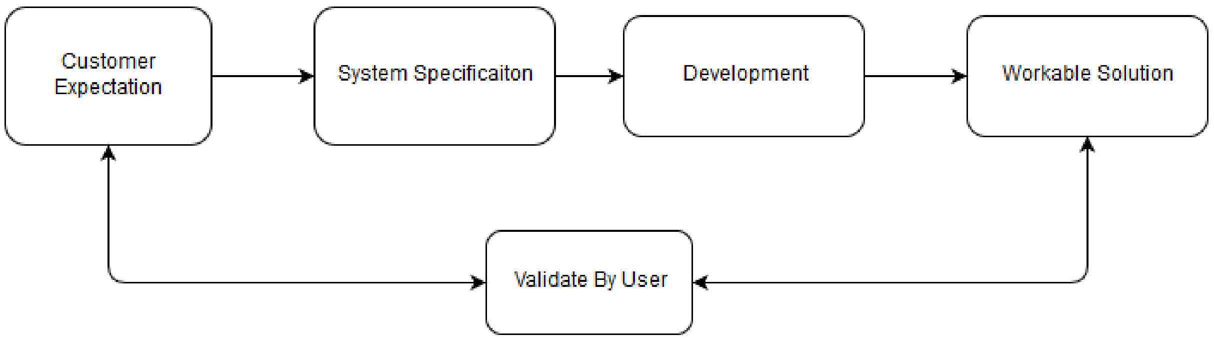


Figure 4.01 verification and validation flow

4.5 System Testing

System Testing is the testing of a complete and fully integrated software product. The bug tracking system is only one element of a larger ERP system. Ultimately a full ERP system should be work as it is after adding this new module (bug tracking system) after the integration. A system test was carried out to verify that all other ERP modules work accordingly (Figure 4.02).

4.5.1 System Testing flow

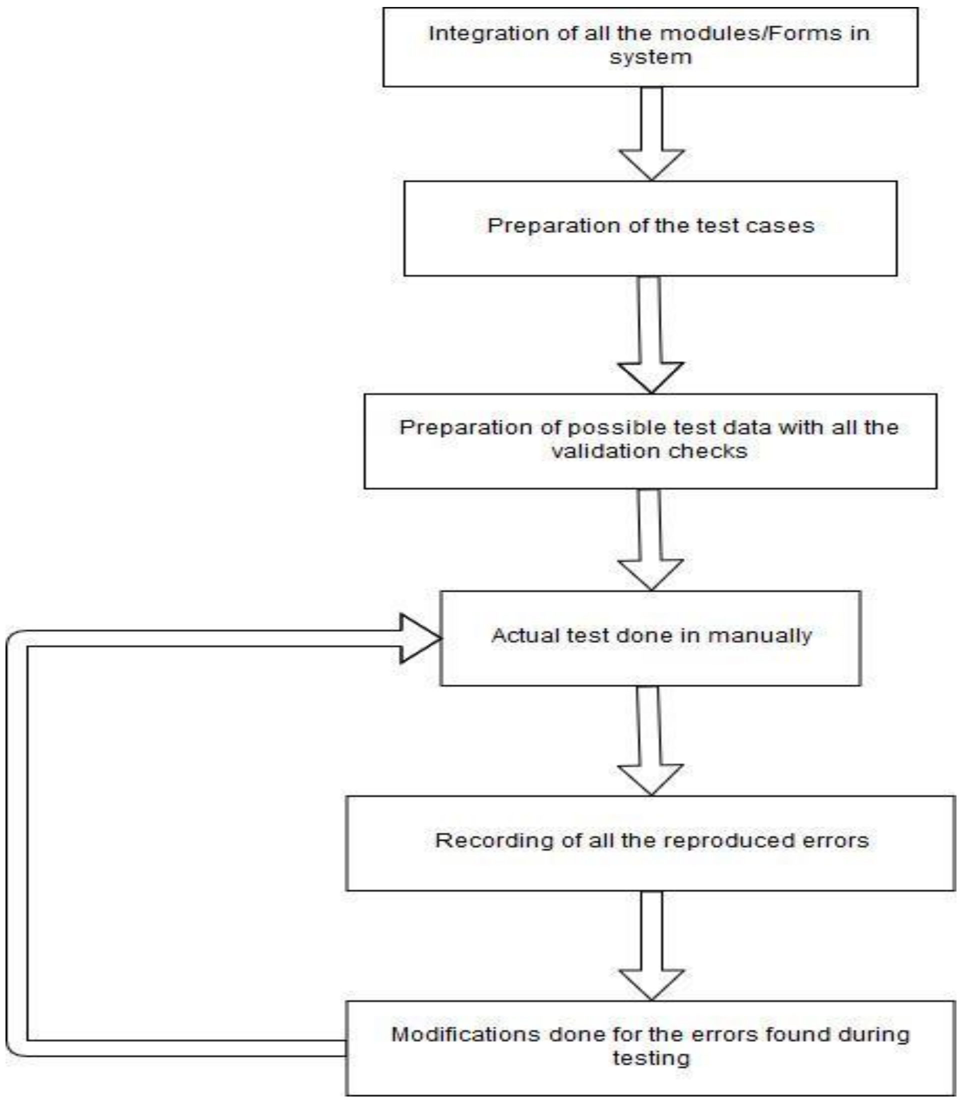

Integration of all the modules/Forms in system
Preparation of the test cases
Preparation of possible test data with all the validation checks
Actual test done in manually
Recording of all the reproduced errors
Modifications done for the errors found during testing

Figure 4.02 System testing steps

Following items were considered in system testing

- The functionality of the entire system as a whole
- The user interface of the system
- Testing the dependent modules together with all the possible test data scripts
- Verification and validation testing
- Testing the reports with all its functionality

After the completion of the system testing the next following phase was the acceptance testing.

4.6 Acceptance testing

- **Alpha testing**

The alpha test is a type of acceptance test; This was done to identify any possible problems/errors before launching the product for ordinary users or the public. Alpha testing is conducted in a corporate environment by internal teams without the participation of development teams, before the launch of external clients. Since the alpha test was done in the last stage of the product before launching it to customers.

- **Beta testing**

The beta test of a product is performed by real external users of the software application with the existing client and is considered a form of acceptance test by the external user.

The beta version of the product has been distributed to a limited number of end-users who already have a good knowledge of our existing ERP system. From this beta version, the tests reduce the risk of product failure and improve the quality of the product through customer validation.

This was the last test done before the product was distributed to customers. The biggest benefit of this beta test was getting direct feedback from customers.

4.7 Usability evaluation experiment with users with a questionnaire

Questionnaire are distributed with the user in order to get the evaluation review. The distributed questioner is in the appendix section

Chapter 5

Conclusion

The results obtained after the evaluation of our test cases and the interrogators conclude that the developed system is a good product. The concept of customer complaint management system is applicable in the field of software engineering and should be used to effectively monitor and investigate defects. Although the size of the data used is small, the results show that 90% of the test cases were completed successfully and that the overall user experience was obtained from the questionnaire, and also shows that users are satisfied with the recently implemented system.

In general, this customer complaint management system is sufficiently capable of meeting most of the proposed system requirements, including proper fault monitoring at all levels. At administrator and administrator levels, it is possible to generate fault reports and defect assignments to affected parties. Developers and project managers can use the software tool at any level to manage the software process based on their need for general defect coverage. It also helps them to focus on a particular type of default report based on usage in the project.

During the development phase of the system, problems were encountered and appropriate solutions were applied to resolve them. Several recommendations for future researchers preceded this in-depth study. The following recommendations are:

- Multiple languages - Currently there is only one language available, which is English. The system can be enhanced to have multiple languages, such as English, Sinhala Tamil and allow the user to choose the preferred language from 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 – by live chat room support user can resolve their usability issues and Q&A directly from the help of support persons because most of the bugs reported by the user are cause users' mistakes.
- 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.
- Create history records – currently, there are no historical records or audit logs. If the user changes the state or comment on a function there is no trace which user has made the change.in future development, it is a mandatory requirement for the system to maintain history and audit logs.
- More analysis reports – The Designed system is having few reports. More analysis reports should be developed for evaluating overall business status.

Appendix section

Following section describe the questioners that have been given to user forget the natural feedback from the user

1. Considering your complete experience with our software, how likely would you be to recommend its use to a friend or colleague?

Very Unsatisfied						Very Satisfied				
0	1	2	3	4	5	6	7	8	9	10

	Very Unsatisfied	Somewhat Satisfied	Neutral	Somewhat Unsatisfied	Very Unsatisfied
2. How user-friendly is our software's interface					
3. What is your overall feeling about the performance of the system					
4. Clarity of Documentation					
5. How useful is for you?					
6. How reliable this software					
7. Is customer support for our software helpful?					
8. How satisfied are you with the customer support team for them?					
9. How easy to Learning and operate the system					

Figure 5.02 user Questionnaire

10. Do you this application is useful to manage bugs
    - Yes
    - No

11. Compare with other existing bug tracking software what are the pros of this product

12. Compare with other existing bug tracking software what are the cons of this product

13. Please put down in your own words areas where we need to improve

14. How visually appealing is our website?

15. Is our pricing strategy clear

The following section shows our Test Cases with their purpose and expected results

TEST CASE	**DESCRIPTION**
TEST CASE ID	001
PURPOSE	Test if the user is able to login successfully.
PRECONDITION	User must be registered already
INPUT	correct username, correct password
EXPECTED RESULT	User must successfully log in to the system

Table A.1 Test Case 1

TEST CASE	**DESCRIPTION**
TEST CASE ID	002
PURPOSE	Test if unregistered users are not able to login to the site
PRECONDITION	
INPUT	incorrect username, incorrect password
EXPECTED RESULT	The proper error must be displayed and prompt to enter login again

Table A.2 Test Case 2

TEST CASE	**DESCRIPTION**
TEST CASE ID	003
PURPOSE	Test with a valid username and empty password such that login must get failed
PRECONDITION	User must be registered already
INPUT	a valid username and empty password
EXPECTED RESULT	The proper error must be displayed and prompt to enter login again

Table A.3 Test Case 3

TEST CASE	**DESCRIPTION**
TEST CASE ID	004
PURPOSE	Check of the password is masked on the screen i.e., the password must be in bullets or asterisks
PRECONDITION	
INPUT	some password (can be a registered/unregistered)
EXPECTED RESULT	The password field should display the characters in asterisks or bullets such that the password is not visible on the screen

Table A.4 Test Case 4

TEST CASE	**DESCRIPTION**
TEST CASE ID	005
PURPOSE	Verify the URL without logging into the site
PRECONDITION	
INPUT	Without username and password
EXPECTED RESULT	the URL should not redirect to a logged-in page but to a logged-out page of the site

Table A.5 Test Case 5

TEST CASE	DESCRIPTION
TEST CASE ID	006
PURPOSE	Verify that the login screen is having the option to enter username and password with submit button and option of forgot password
PRECONDITION	
INPUT	
EXPECTED RESULT	The login screen should have username, password, and submit button

Table A.06 Test Case 6

TEST CASE	DESCRIPTION
TEST CASE ID	007
PURPOSE	Verify that the password is in encrypted form when entered
PRECONDITION	Registered user
INPUT	
EXPECTED RESULT	Check from database password is encrypted

Table A.07 Test Case 7

TEST CASE	DESCRIPTION
TEST CASE ID	008
PURPOSE	Verify if SQL Injection attacks work on the login page
PRECONDITION	
INPUT	
EXPECTED RESULT	Check SQL Injection is possible

Table A.08 Test Case 8

References

[1] L. Kierczak, "survicate.com," survicate, 12 06 2018. [Online]. Available: https://survicate.com/customer-satisfaction/importance-customer-satisfaction/. [Accessed 15 05 2019].

[2] "15 Best Bug Tracking Software: Top Defect/Issue Tracking Tools of 2018," softwaretestinghelp, 18 Nov 2018. [Online]. Available: https://www.softwaretestinghelp.com/popular-bug-tracking-software/. [Accessed 28 September 2018].

[3] "bugzilla," bugzilla, 2018 February 2016. [Online]. Available: https://www.bugzilla.org/.

[4] "bugzilla," financesonline, [Online]. Available: https://reviews.financesonline.com/p/bugzilla/. [Accessed 18 03 2019].

[5] "Bugzilla REVIEW," financesonline, 14 03 2019. [Online]. Available: https://reviews.financesonline.com/p/bugzilla/.

[6] "User Review: "Bugzilla for Bug Hunters"," trustradius, 29 06 2016. [Online]. Available: https://www.trustradius.com/products/bugzilla/reviews/pros-and-cons?cg=small.

[7] "Jira," atlassian, 18 03 2019. [Online]. Available: https://www.atlassian.com/software/jira.

[8] "Jira REVIEW," financesonline, [Online]. Available: https://reviews.financesonline.com/p/jira/.

[9] "JIRA Software review," trustradius, 22 July 2016. [Online]. Available: https://www.trustradius.com/products/jira-software/reviews/pros-and-cons?f=50.

[10] "JIRA Software," getapp, 20 04 2019. [Online]. Available: https://www.getapp.com/project-management-planning-software/a/jira/reviews/.

[11] "5 reasons NOT to choose Atlassian JIRA for agile projects," linkedin, 20 10 2015. [Online]. Available: https://www.linkedin.com/pulse/5-reasons-choose-atlassian-jira-agile-projects-piotr-pracuch.

[12] K. J. a. B. R. Alan Page, How We Test Software at Microsoft, 2008.

[13] I. C. N. Serrano, "Bugzilla, ITracker, and other bug trackers," *Bugzilla, ITracker, and other bug trackers,* pp. 11 - 13, 2005.

[14] "Comparison of Extraction Methods for Bug Tracking System Analysi," *International Conference on Information Science and Applications (ICISA),* pp. 24-26, 2013.

[15] L. B. Ruchika Malhotra, "A defect tracking tool for open source software," *International Conference on Information Science and Applications (ICISA),* pp. 7-9, 2017.

[16] "Bug Tracking and Reporting System," *International Journal of Soft Computing and Engineering (IJSCE),* vol. 3, pp. 42-45, 2013.

[17] "15 Best Bug Tracking Software: Top Defect/Issue Tracking Tools of 2019," 23 April 2019. [Online]. Available: https://www.softwaretestinghelp.com/popular-bug-tracking-software/. [Accessed 23 April 2019].