

Project Management Plan

Strix - Multi-tenant Issue Tracking and Reporting System

Faculty of Information Technology
University of Moratuwa
2020

1. Introduction

The Strix Bug Management System is a user friendly and simple open-source bug tracking software. The purpose of developing this software is to increase the ease of non-technical user engagement with the project team by focusing on the user friendliness in contrast to technical facilities. After addressing the aforementioned customer engagement issue, we also implement essential facilities for Agile Scrum based bug management system and reporting system to assist developers, QAs, and project managers.

2. Project Management Approach

The Strix Bug Management System will be developed according to the agile software development process model as it has a high probability of changes in requirements. The system was divided into five modules by the client. The team members are responsible for adding tasks for the 30-day sprint according to their ability and availability and to complete within the period. Work of the previous day, plans of the current day, and issues each member faced during the previous day will be discussed and resolved during daily team meetings.

The role and the responsibilities of each person associated with the project are mentioned in the following table.

Name	Role	Responsibility
Dr. Priyanga D. Thalagala	Project supervisor	Project supervision
Mr. Manujith Pallewatta	Project mentor	Project review
	Client	Provide changes in requirements
M.D.N. Chandrasiri	Project leader	Ensure smooth operation Overseeing member work
	Team member	Accurate delivery of tasks Timely delivery of tasks Maintain meeting minutes
E.A.Y.R. Edirisinghe	Team member	
W.P.C.P. Pathirana	Team member	
W.A.D. Sandarekha	Team member	
I.K.G.D.S.N. Senanayake	Team member	

3. Project Scope

3.1 Aim and Objectives

3.1.1 Aim

The aim of this project is to develop a system to bring all user roles which are the customer, developer, QA, and the manager to one platform to solve the issues/bugs regarding a selected product with the use of the latest technologies in the industry.

- Reactjs
- Django
- Django Rest
- Postgresql

3.1.2 Objectives

- A critical review of bug tracking systems
- In-depth study of latest developments in technologies
- Design and create wireframes for the interfaces
- Develop separate systems for each user roles
- Evaluation of the proposed solution
- Preparation of final documentation

3.2 Product Functions

The bug management system has been divided into five submodules according to their functionalities.

1. Authentication and Authorization
2. Bug Capture Log (BCL)
3. Bug Management System (BMS)
4. Bug Solution Pool (BSP)
5. Bug Reports (BR)

Each sub-module has its unique functionalities and different access points for users.

3.2.1 Authentication and Authorization

The Authentication and Authorization sub-module can be specified as the initiation module of the system. Strix Bug Management System follows Role-Based Access Control (RBAC) method.

Following user stories fall under this sub-module.

- **Login**
 - As an administrator, I want to log into the admin home, so that I can navigate to every other function.
 - As an administrator, I want to reset my password, so that I can get a new password.
 - As a Manager, I want to log into the system, so that I can manage the system.
 - As a Manager, I want to reset my password, so that I can get a new password.
 - As a QA, I want to log into the system, so that I can review issues.
 - As a QA, I want to reset my password, so that I can get a new password.
 - As a Developer, I want to log into the system, so that I can solve issues.
 - As a Developer, I want to reset my password, so that I can get a new password.
 - As a Customer, I want to log into the system, so that I can report an issue.
 - As a Customer, I want to reset my password, so that I can get a new password.
- **User manipulation**
 - As an administrator, I want to create external user accounts, so that I can add newly added external users to the system.
 - As an administrator, I want to create internal user accounts, so that I can add newly added internal users to the system.
 - As an administrator, I want to update external user accounts, so that they have accurate details.
 - As an administrator, I want to update internal user accounts, so that they have accurate details.
- **Project manipulation**
 - As an administrator, I want to create projects, so that I can add newly created projects to the system.
 - As an administrator, I want to assign users to a particular project, so that I can provide access to the users to a particular project.
 - As an administrator, I want to update project details, so that they have accurate details.

3.2.2 Bug Capture Log

BCL is the interface where external users (customers) can engage with the system. Customers who log into the system through the Authentication and Authorization sub module will be directed to the BCL module.

Following user stories fall under this sub module.

- **BCL home**
 - As a customer, I want to view related projects to me, so I can communicate with developers who work on my projects.
- **BCL backlog**
 - As a customer, I want to view issues I have previously filed, so that I can check their progress.
- **Issue form**
 - As a customer, I want to create a new issue, so that I can inform the developers about it.
- **File uploading**
 - As a customer, I want to upload files, so that the developers can get a better idea about the issue.
- **Screen capturing**
 - As a customer, I want to record my screen, so that the developers can get a better idea about the issue.
- **Ticket View**
 - As a customer, I want to comment about existing bugs related to me.

3.2. 3 Bug Management System

This sub module will be only available for internal users (managers, QAs, and developers). Internal users who log into the system through the Authentication and Authorization sub module will be directed to the BMS module.

Issues created by customers in the BCL module will be visible in the BMS issue backlog which is a pool of all the bugs belonging to that particular project.

It should be noted that a sprint will consist of a Kanban board with four lanes. The four lanes will be,

- Open
- In progress
- Review
- Completed

Following user stories fall under this sub module.

- **BMS home**
 - As a manager, I want to view projects that I am a part of, so that I can access their issues.
 - As a QA, I want to view projects that I am a part of, so that I can access their issues.
 - As a developer, I want to view projects that I am a part of, so that I can access their issues.
- **Issue backlog**
 - As a manager, I want to view the issue backlog, so that I can access issues.
 - As a manager, I want to sort and filter issues in the backlog, so that I can access issues that match the conditions.
 - As a QA, I want to view the issue backlog, so that I can access issues.
 - As a QA, I want to change the tags of the bug if necessary so that developers get the correct idea of the issue.
 - As a QA, I want to add comments to the issue, so that I can get a better understanding of the issue.
 - As a developer, I want to view the Issue backlog, so that I can access issues.
 - As a developer, I want to add comments to issues, so that I can get a better understanding of the issue.
 - As a developer, I want to attach files to the issue, so that customers can view them.
 - As a developer, I want to add time spent on an issue, so that the company can track my working hours.
 - As a developer, I want to mark issues for Bug Solution Pool, so that the manager can approve or reject them.
- **Sprint backlog**
 - As a manager, I want to view the sprint backlog, so that I can access sprints.
 - As a manager, I want to create sprints and customize their due date so that I can add issues to them.
 - As a manager, I want to add issues to the sprint so that developers and QAs can work on them.
 - As a manager, I want to pin sprints to the sidebar, so that I can access certain sprints easily.
 - As a manager, I want to end a sprint, so that I can update the status of the sprint.
 - As a QA, I want to view the sprint backlog, so that I can access sprints.
 - As a developer, I want to view the sprint backlog, so that I can access sprints.

- **Kanban board**

- As a QA, I want to review issues that are in the “Review” lane of the kanban board, so that I can make decisions regarding which lane to move the issue into.
- As a QA, I want to move issues from the “Review” lane to the “Done” lane, so that the issue is marked as completed.
- As a QA, I want to move issues from the “Done” lane to the “Review” lane, so that I can re-review the issues that I mistakenly identified as completed.
- As a QA, I want to move issues from the “Review” lane to the “In progress” lane, so that developers can work on them again.
- As a developer, I want to move issues from the “Open” lane to the “In progress” lane, so that I can work on it.
- As a developer, I want to move issues from the “In progress” lane to the “Open” lane, so that someone else can work on it.
- As a developer, I want to move issues from the “In progress” lane to the “Review” lane, so that a QA can review it.
- As a developer, I want to move issues from the “Review” lane to the “In progress” lane, so that I can rework on issues that I mistakenly moved to the “Review” lane.

3.2.4 Bug Solution Pool

Bug Solution Pool is a platform where resolved bugs are referenced as a knowledge resource. Managers, QAs, developers are allowed to use this sub module.

Following user stories fall under this sub module.

- As a manager, I want to approve or reject issues that developers have marked for the Bug Solution Pool so that developers and QAs can view them in the Bug Solution Pool.
- As a manager, I want to view issues and previously added comments, so that I can understand the issue.
- As a manager, I want to search for issues, so that I can access the necessary issues.
- As a QA, I want to view issues and previously added comments, so that I can understand the issue.
- As a QA, I want to add new comments.
- As a QA, I want to search issues, so that I can access the necessary issues.
- As a developer, I want to view issues and previously added comments, so that I can understand the issue.
- As a developer, I want to add new comments.
- As a developer, I want to search for issues, so that I can access the necessary issues.

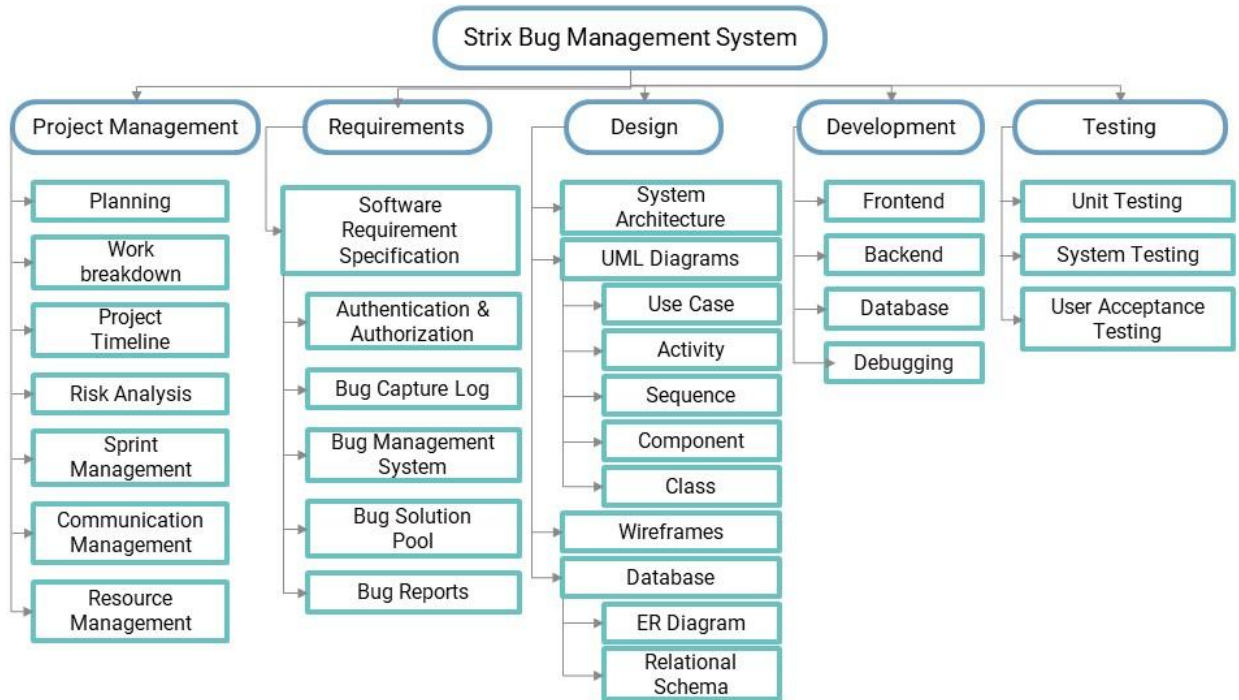
3.2.5 Bug Reports

Bug Reports sub module exists to provide graphical representation and an overall summary of the issues, users, sprints, and other details. The following reports are supported in this sub module.

- As a manager, I want to view the “Developer timesheet”, so that I can get information about issues worked on by a particular developer and the time taken to resolve each of them.
- As a manager, I want to view a stacked bar chart from the developers' perspective, so that I can see total bugs assigned, total resolved issues, and total progress along with the month.
- As a manager, I want to view developer performance in a pie chart, so that I can see the total assigned, resolved, and in-progress tickets.
- As a manager, I want to view developer average effort in a bar chart, so that I can see the average hours spent on an issue along with the month.
- As a manager, I want to view the “Project timesheet” as a table, so that I can see progress on multiple projects.
- As a manager, I want to view a stacked bar chart on projects, so that I can see total bugs assigned, total resolved issues, and total in progress.
- As a manager, I want to view the “Sprint summary”, so that I can see details about the sprint.
- As a manager, I want to view the table “Monthly bug summary”, so that I can see issues grouped by the month of reporting.
- As a manager, I want to view a bar chart of the total number of bugs reported, so that I can see how many bugs have been reported in each month.
- As a manager, I want to view a stacked bar chart on bugs, so that I can see total bugs assigned, total resolved issues, and total in progress along with the month.
- As a manager, I want to view the full bug summary, so that I can see total bugs reported, total projects with active bugs, and bug completion as a percentage.

4. Project Schedule

4.1 Work Breakdown Structure



4.2 Gantt Chart

The Gantt Chart of the project can be viewed using the following link.

[Gantt Chart](#)

5. Communication Management Plan

The communication management plan works as a guide for all the communications throughout the project duration. The project leader ensures the effective communication of the project. The following communication matrix describes the communication requirement, what information to communicate, method of communication, frequency of communication and participants. Any issues, concerns or updates will be communicated to the project leader.

Communication Type	Description	Frequency	Format	Participants
Project review	Review of project progress	Twice a month	Virtual meeting	Project supervisor Team members
Project review	Review of the project progress Clarification of any requirement changes	As needed	Virtual meeting	Project mentor Team members
Project team meeting	Review of project progress Technical issues Next steps	Daily	Informal communication	Team members

6. Risk Management Plan

	Risk Type	Risks	Prevention and Avoiding Techniques
Backend Risks	Security Risks	Injection Flaws	Uses built-in Django ORM where developers do not need to create raw SQL. Moreover, writing good error handling can improve the protection.
		Broken authentication	Encourage users to adopt a good password policy. Uses a hash algorithm to hash user passwords.
		Broken access control	Forces accessing different URL endpoints by using Django permission classes.
		Insecure Deserialization	Every user data is going to be validated before returning a response at every time.
	API Risks	Improper API build	Follows a proper API structure to build Rest API.
		Improper Coding	Avoiding unwanted loops and modularizing code snippets to make it reusable.
Frontend Risks	Security Risks	Client XSS (Cross-Site Scripting)	React automatically escapes variables for developers. It prevents XSS injection via string HTML with malicious Javascript. Naturally, inputs are sanitized along with this.
		Broken Access Control	By using react private routers which invoke only when the user was validated against the backend. This overcomes mentioned security pitfalls.

		Arbitrary Code Execution	One of the robust approaches to tackling arbitrary code execution is to ensure that the application only reads the tokens that have been stored previously during the development. After this, we can ensure that the system only produces relevant headers by authenticating the request by making one to the server.
		Invalid use of HTTP methods	Follows a proper way of using HTTP methods. Ex: Use POST when submitting a form
Database Risks	Security Risks	SQL Injection	Database items will not be accessed using SQL statements rather use a built-in ORM in Django.
		Natural Failures	Cloud-based storage is used to deploy databases even in the development stage.

Above mentioned risks are identified as up to now. Risks can occur at any time while we are developing the system. Since we are using agile as our software development methodology, risks can be undertaken from time to time.

7. Resource Management

Resource	Status
Team	<p>There are five second-year undergraduates who are working on the project.</p> <p>Three out of five members have experience in React development.</p> <p>One member has experience in the Django framework.</p>
Time	<p>Each member will put 20 hours of effort into each month considering learning, development, and testing aspects.</p> <p>The project development and testing will go on for five months.</p>
Equipment	<p>The project will be done on the personal computers of team members.</p>
Storage	<p>During the development phase, the database will be hosted on ElephantSQL.</p> <p>At the implementation phase, the database will be hosted on the client's cloud storage.</p>
Monetary	<p>The project will be done for academic purposes and thus no payment will be made.</p>