Bug Tracker Implementation
Minimum Viable Product
Technical Documentation

Alexandru Manafu

1 Introduction

This document defines a generic Bug Tracker and describes the process of implementing such an application that focuses only on the core features.

We start off with a simple set of questions:

- What is a bug tracker?
- What are its uses?
- What are the most significant features?

2 Overview

A bug tracker is in essence a to-do list that divides tasks (such as fixing bugs) into categories and assigns a priority to each task, by doing so the amount of time needed for task completion should be reduced while application quality should increase.

Typically bug trackers are a must for team projects. Once an item is changed or added the right people should get notified.

Core features of my implementation:

- Data visualization (Kanban-style board)
- Multiple Account types with different permissions (project manager, developer)
- Task priority

3 Scope and Goals

Each project manager type account can create projects and developers will join projects by inputting a code somewhere.

This means that the possibility of every type of account becoming a manager if they create their own project is out of this project version's scope.

Each project will have some default tables, in a more complex version of this project each manager would create all projects from scratch by making all the tables.

The goal is to have a web-based application with an easy to use UI.

3.1 Risk management

Implementation for regular user type account will be delayed until after Developers are able to join projects, if there is not enough time the regular user type account will be abandoned.

If the implementation of multiple projects for each project manager fails then each project manager type account will have one project.

4 Website contents and technologies

The main theme of the website is going to be sort-of dark, buttons and some other elements will be colored nicely.

The about page(accessed via the About button) will briefly describe the website.

The getting started page(accessed via its respective button) will contain some information and a link to the user guide.

The log-in button will open a drop down menu that will allow a user to log in or if the user clicks another button inside the drop down menu, the user will be redirected to a register page. Most buttons on the website open a drop down menu of some sort.

4.1 Error Checking and Permissions

Error checking is provided and is quite extensive, for any form no inputs can be empty and more specifically for the login if the user types in a wrong user-name or password an error will pop up. For the register page the following cases will show an error :

- The user-name exists or the user-name contains more than just numbers and letters.
- The email is invalid.
- The password is shorter than 8 characters or the two passwords don't match.

If the user tries to access any of the script files directly the user is redirected to the index page. In the project pages the following will not be possible:

- The user tries to access the project pages without being logged in.
- The user tries to access a project that he did not join properly.
- The user tries to select issues that do not belong to the current selected project.

4.2 Technologies

The top menu is a PHP file that is included in every other page.

When the user logs in some session variables are set that contain the user type, user id, etc. In the projects page after the user logs in there is yet another button for joining and creating a new project this is implemented as a form with POST which in turn runs some scripts if the user actually send the form properly. This process is also used extensively in issue modification.

A more interesting topic is the displaying of existing projects from the projects table. The logic behind the implementation is:

- First create an object like element that would represent a project in this case an icon with the projects title beneath was chosen.
- The next step would be to fetch all the entries that need to be displayed in this case a function is used to extract all projects that are owned by the user (if the user is a manager) or joined by the user (if the user is a developer).
- Finally by using the previous result (an associative array) we can display the entries, it was decided that they would actually be links with GET methods inserted in them.

In the project page once the user selects a project the first thing to notice is the 5 columns, on each column a function is used to extract and display the issues belonging to the selected project while also belonging to the column (Backlog, To Do, ...).

Each displayed issue is once again a link containing 2 GET methods: current project code and selected issue id.

Most buttons in the project page belong to a common CSS class such that if the user click on one of them it hides all other drop down windows and activates only the window belonging to the button pressed.

Buttons that modify issues contain 2 hidden inputs with the project code and the current selected issue an they work like this: once pressed a confirmation window or a from appears and upon clicking Yes, No or Apply the form is send and some other functions are called inside the script then the user is redirected to the same project and the previous selected issue (is it is the case) this is done with the 2 hidden inputs.

4.3 Database Structure

An empty database called 'bugtracker' must be created before the application goes online. A function that can create the default tables is provided.

Initially with no projects created the database should contain the tables:

- users (columns user type, user name, etc)
- projects (columns: project code, project name and project owner)
- developers (columns: developer user-name, project code)
- issues (columns: id, place, title, details, priority, createdBy, developedBy, project code)

Then after a new project is created by a manager an entry is added into the projects table.

When a new issue is created by a user an entry is added to the issues table.

Each project will have the following rules, when you create an empty project you automatically get some default tables with the 5 of them named Back-log, To-do, In Progress, Testing, Completed.

For further info on issue modification read the user guide beginning from section 2.