System Design
Bug Tracker

May 30, 2021

Contents

1 Introduction

This document has been created in order to provide a more detailed insight for the project. Also these documents should focus on parts of the system not yet implemented such that it helps the developers better shape those parts.

The initial (or previous) goal of the system was to provide a bug tracker application in which the manipulation of issues does not depend entirely on project managers. Now the goal is to add more functionality.

Features present in the first version:

- The **Register** and **log-in** system.
- The **Projects** page with the create, join and select project functionality.
- The **Kanban** page with the features: Select Issue, Move issue, Delete Issue, Edit Issue.

For full context compare these two repositories:

- github.com/AlexandruManafu/Bug-Tracker-V1
- github.com/AlexandruManafu/Bug-Tracker-Final

2 System Overview

The key goals of the system are:

- Provide more tools for project managers and to separate the project management from issue management
- Mobile compatibility achieved with a new format for displaying issues and responsive design.
- Notifications are intended to provide some automatic messages and will be used mostly when 2 actors are involved

For each addition or change a top down approach is used, this means that the user interface is the first thing that is changed.

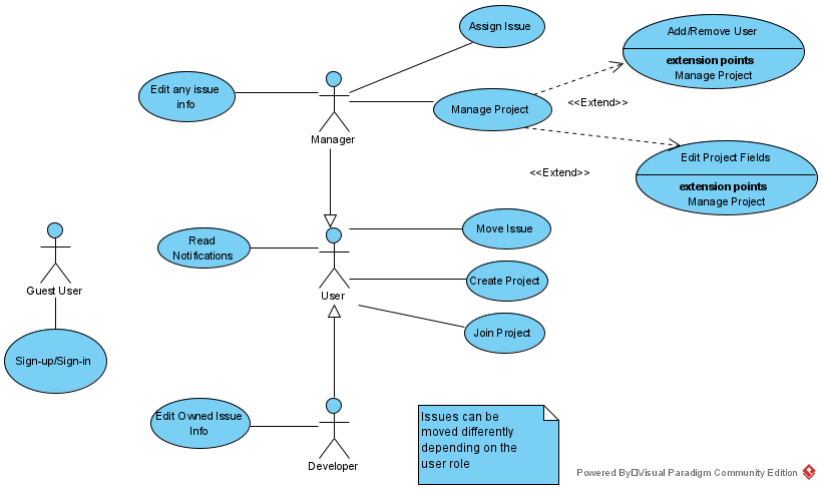
The system is a dynamic web application in which the users are associated with projects and where issues are also associated with projects. The role of the user determines a sort of permission system, for example only project managers can add users in the project and they also have access to the project management page. The users may have different roles for different projects.

Most functionality either creates entries in a database table, displays them or changes them.

Info entered in forms that trigger actions with the database are checked such that they are not empty and all functions dealing with the database use placeholders such that SQL injections should not be possible.

Users will get redirected if they try to access script files or try to access projects that they are not associated with, moreover issues are only accessible from the projects they are created in.

The use case diagram:



3 Design Constraints

A limitation of the system is that each project may have one project manager.

What the system does not try to achieve:

- File management (like Git)
- User Messaging (like a chat client)
- Email Notifications

4 Roles and Responsibilities

Roles:

- Project Lead - Alexandru Manafu
- Developer - Marius Bijan

Resposibility list:

- Project Documentation - Project Lead
- Merging the code, or simple assistance - Project Lead
- Development of Responsive Design - Developer
- Development of some new pages (front-end) - Developer
- Adding functionality (back-end) - Project Lead

Unpredictable Issues also should be the responsibility of the developer but this does not mean everything that goes wrong should be fixed by the developer.

5 System Architecture

Only base languages are used to develop the system. Most of the code is procedural PHP.

Technologies and what they do:

- HTML is used to create basic elements
- CSS is used to stylize elements
- PHP is used as a bridge between the User Interface and the database, and is also used for the dynamic parts and most actions.
- Javascript is used mainly to toggle some elements
- MySQL is used as the database part

6 Database Design

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

The database tables and their roles:

- users - used for signing in the application
- projects - for storing projects, selecting a project and other project management actions
- developers - makes it possible for users to join projects
- issues - stores all information about all issues and what project they belong to, makes it possible to manipulate issues by changing fields
- notifications - stores all notifications, makes it possible for users to receive notifications

The database tables and their attributes:

- users(id,name,email,password)
- projects(id,code,name,description,owner)
- developers(name,projectCode)
- issues(id,...,projectCode) where ... are all the issues descriptive fields
- notifications(id,content)

7 Software Detailed Design

The basic idea of any feature of the system is that the user selects something then one of these things will happen:

- Another page loads: the triggers in this case are links and this is how the navigation menu, project selection and issue selection work
- An element is toggled: the triggers are buttons outside forms, this is how elements are hidden and only viewed if needed
- Some script is run: the triggers are buttons within forms, the script does what its supposed to do and the user is redirected back to the initial page or another page.
  Most if not all actions that change or create something is done with this.

The menu is implemented as a list of links with HTML, more options appear if the user is logged in and this, like all dynamic parts it is done in PHP.

In the Projects page, the buttons for project creation will display a hidden form once clicked.

The projects themselves are actually links that will redirect the user to another page. The list of projects is dynamic and to display it the system takes everything from the database.

8 System security and Integrity controls

We will consider 3 levels of importance with 3 being the lowest.

- Level 1 that is database security and user security.
- Level 2 is making sure that script files are not accessible
- Level 3 is making sure that users cannot access some other project or project's issue

Measures taken to ensure security and integrity

- SQL injections should not be possible because everything uses placeholders
- Passwords are stored in a hashed form
- Forms will not take empty inputs
- Script files redirect users if they try to access them directly
- The project join code is not visible, only the project manager can get it
- Users are redirected if they try to access a project that they did not join
- Issues that do not belong to the selected project will not be displayed and cannot be targeted

9 Project References

The Requirements document contains info about functionality that is about to be implemented.

The Planning document specifies a list of features to be implemented and a a GANTT chart that is supposed to approximate the time needed.

The User Manual document specifies features of the system such that anyone can understand what the system can do and to provide some support in using the system.