System Requirements Specification

For

BUG REPORTING SYSTEM

Prepared By:-

Ananya Adiga (PES2UG20CS043)
Aanchal Agarwal (PES2UG20CS005)
Anirudh Chakravarty Kumar (PES2UG20CS049)
Adarsh Liju Abraham (PES2UG20CS017)

TEAM X-4B

Version 2.0
September, 2022

Table of Contents - Adarsh Liju Abraham

Revision History

Name	Date	Description of version	Version
Team X-4B	11-09-22	The final version of the SRS document has been drafted with all the requirements being incorporated into the document.	2.0

1. Introduction

1.1 Purpose

This document specifies the software requirements for a Bug Reporting System.

This document is intended to

- Explain the purpose and features of the software
- The feasibility of the project
- How the product would respond to various bugs

The document gives a clear and concise description about the hardware and software interfaces, constraints, features that meet the user stories.

1.2 Product Scope

A bug tracking software keeps all issues relating to your bugs in the same place and ensures that they are fixed appropriately.

- One of the primary objectives of the bug tracking tool is to reduce the cost of the development.
- Allow companies track all the bugs in a centralized location to know the bugs reported,resolved and how complex it was.
- Help bugs to be solved within a short turnaround time thereby preventing losses occuring due to bugs in software.
- Reward certain individuals who contribute in fixing the bugs and keep a track of severe ones.

The points mentioned above would help developers and testers enhance their team's productivity ,resolve bugs without disruption, and deliver high-quality products quickly and efficiently.It has tools for quicker bug resolution, which allows the development team to avoid product release delays.As a result it leads to reduction in production cost and a higher return on investment for the product company.

1.3 Glossary

This subsection contains definitions of all the terms, acronyms, and abbreviations used in the document. Terms and concepts from the application domain are defined.

1	User	User is an individual who is on the client side and reports the bugs he is facing
2	Member	Member works on the server side and picks up a reported bug and resolves them
3	Ticket number	A unique number will be generated to every bug reported by an individual, with which the user can check the issue status and the solution anytime
4	Admin	Admin works on the server side and once he logs in checks the bugs reported and assigns them to the members
5	SRS	SRS stands for Software Requirement Specification. It is a document that completely describes all of the functions of a proposed system and the constraints under which it must operate.

6	Constraint	Issues that a software faces w.r.t to time or design which doesn't let the system achieve its goals

1.4 References

- https://ieeexplore.ieee.org/document/5070993
- https://www.ibm.com/in-en/topics/bug-tracking
- https://ieeexplore.ieee.org/document/4639
- https://www.researchgate.net/figure/Hardware-and-Software-Requirement_tbl1_49587841

1.5 Overview

The first chapter, that is the Introduction section of the document,is intended to introduce the reader to the product,**insecTo_destrucTo**.

The second chapter, Overall Description section of SRS v2.0 document provides an overview of the overall functionality of the product. It describes about the hardware and software interfaces.

The third chapter, Specific Requirements section, of SRS v2.0 document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

The second and the third chapter of the document describe the same software product, but are intended for different audiences and thus use different languages.

2. General Description

A bug refers to any defect in the design, coding, and features of a product that can cause it to malfunction or deliver incorrect results in a product. The process of identifying these defects in a product is known as bug tracking. A bug tracking software is an essential tool for managing the process of finding, tracking, and fixing bugs. Not only does it ensure that you deliver a high-quality near-perfect product, but it also enhances the productivity of the development team throughout the product development process.

2.1 Product Perspective

The insecTo_destrucTo Bug Tracking system is a virtually self contained bug tracking system; however, it will require users to have access to a web browser on their workstation computer. This means that the users of the system do not need to invest in any other software to get the most out of the software system as any Windows based PC comes installed with a web browser, and any non Windows machine can use Google Chrome , FireFox or other freeware browsers. This feature is built into the software and does not require any other software to function.

User interface

The user interface is user friendly and robust, one can easily report the bugs they are facing and login later using the ticket number generated to them to check their issue status.
The status indicates the current progress of the issue reported-

- Open- It is available to be resolved by any of the members(preferably Testers).
- Resolved- The issue is resolved
- Closed- The issue is verified and closed
- Deleted - the issue is deleted to focus on important ones.

Hardware interfaces

The bug tracking software demands just the basic hardware interfaces like -a processor, .about 4GB memory, 1TB hard disk space and other required computer peripherals.

Software interfaces

The server side requirements are an operating system, a programming tool, MySQL database server, apache web server.

The client side requires any internet browser to access the software.

System interface

The software will be designed to work on any version of Windows, Linux and Mac platforms. The software is completely web based and runs on popular web browsers namely Firefox, Google Chrome,Microsoft Edge. These web browsers are preferred since they support HTML , Javascript.

2.1.5 Design Constraints-

- Operates on any platform i.e. the user can access the software on any operating system and on a computer system.
- The user can communicate only in English, the software is not multi-lingual

2.2 Product Functions

The functional & non-functional requirements from **insecTo_destrucTo** have been analyzed and all issues and ambiguities resolved. In some cases, the resolution is a deviation from the requirements supplied by PES(client). This document is for review by PES project management & marketing, and the changes to the spec should be analyzed for acceptability to PES.

PES wanted a robust,fast and responsive system for reporting bugs in the enterprise world. This said framework should support multiple departments and users at the same time.
The specific functional and nonfunctional requirements provided have been segregated in the sections below in section 3. The final requirements will be made after the requirement analysis.

2.3 User Characteristics

The Admin can log in to the app and enter the details of the project and staff members, view the complaints of users, assign tasks to the members, track the bug resolution status, and send messages to users. The staff can log in to the app to view bug complaints assigned to them and start working on them. If a member is unable to resolve a bug issue, they can forward the problem to another member. As for the users, they can view the progress status of their bug complaints.
Users can also use the ticket number generated during the submission of their complaint and view the solution plan details for the same. In this way, the bug tracking application keeps the process of monitoring and resolving software bugs transparent and seamless.

2.4 Constraints

There are a number of constraints which the system must abide by during development. The system must be developed within their bounds. These constraints dictate a number of the functional and nonfunctional requirements specified by this document. Others are because of a requirement specified to us by our customer. All are important to be aware of during the implementation of the software system.

- System is to be developed for distributed use as a web application. This will limit the ability for real time updates to the system.
- System is to be developed in PHP through XAMPP and HTML pages.
- Data must be stored in a relational database for quick queries and storage.
- The styling and template design is done using CSS and Tailwind.
- The Database Used is SQLite which is a fast embedded database
- Passwords are sent and stored in encrypted form.
- Some users are authorized users while some are non-authorized users. Non-authorized users can not see other user's preference and exclusion sets.
- System must be robust enough to handle multiple routes and queries.
- System must be able to access data faster even when the database is filled

2.5 Assumptions and Dependencies

- System will be installed on a machine running Windows/Linux operating system, XAMPP and any Modern Web Browser.
- The administrator also does not interfere in the posting and creation of bug issues.
- The system requires good internet connection and secure protocols to prevent mass data leaks
- Data must be encrypted and SQL queries must not contain problems during execution
- Right now our assumption is that all departments can read the bug issues and comment on them based on their experience.

2.6 Apportioning of Requirements

insecTo_destrucTo fully intends to ship a fully operational and complete system with its first release version. However there is some functionality that could be released at a future date if needed.

The primary functionality that could be held back for future release is the Email Notification feature. The ability to send out emails to users when a new invitation is ready to be viewed in the system for acceptance or decline is a feature which will make the system far more user friendly in the fast paced business world. It is however a feature that the system can be used without and still function in its original scope and requirement.

3. Specific Requirements

This section specifies the detailed requirements which the system shall meet.

3.1 External Interfaces

This section specifies the user interfaces to the system. All user interfacing is done through a web UI. The web pages are shown below.

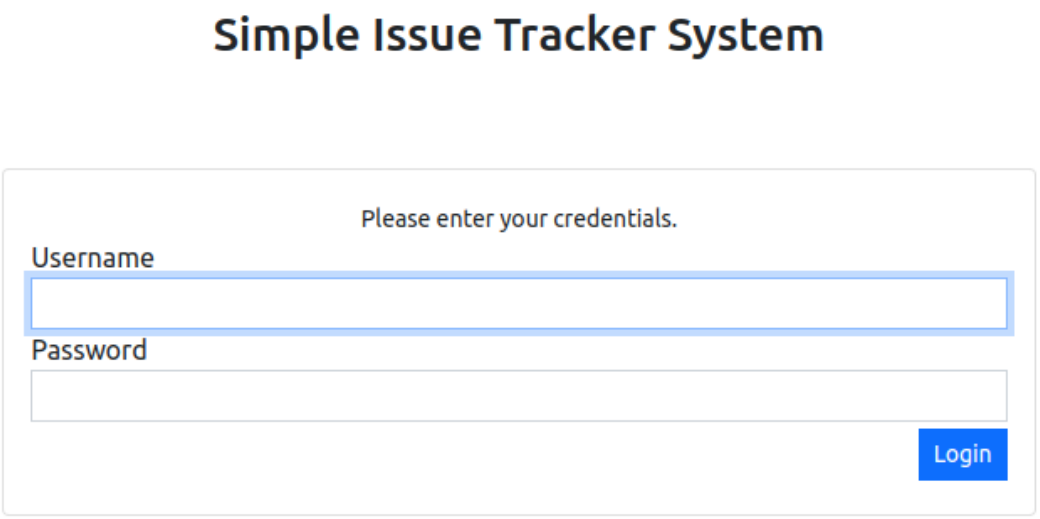


Figure 1: Login Page
The user performs the log in functionality which addresses the non functional requirement of security.

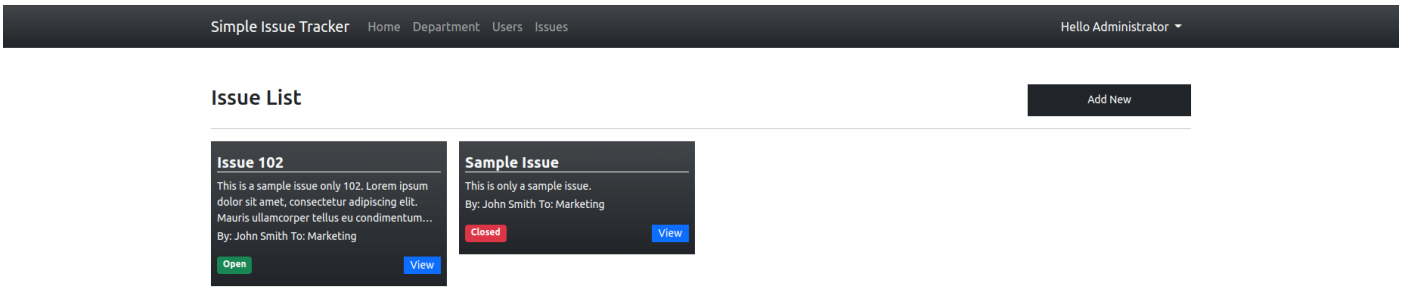


Figure 1: Home page

The home page is where the user arrives after logging onto the system. The issue list in the homepage shows the list of current issues which have to be addressed and the current status of a particular issue and a view button to view the issue. On viewing the issue we can check the details of the specific issues to all the departments.

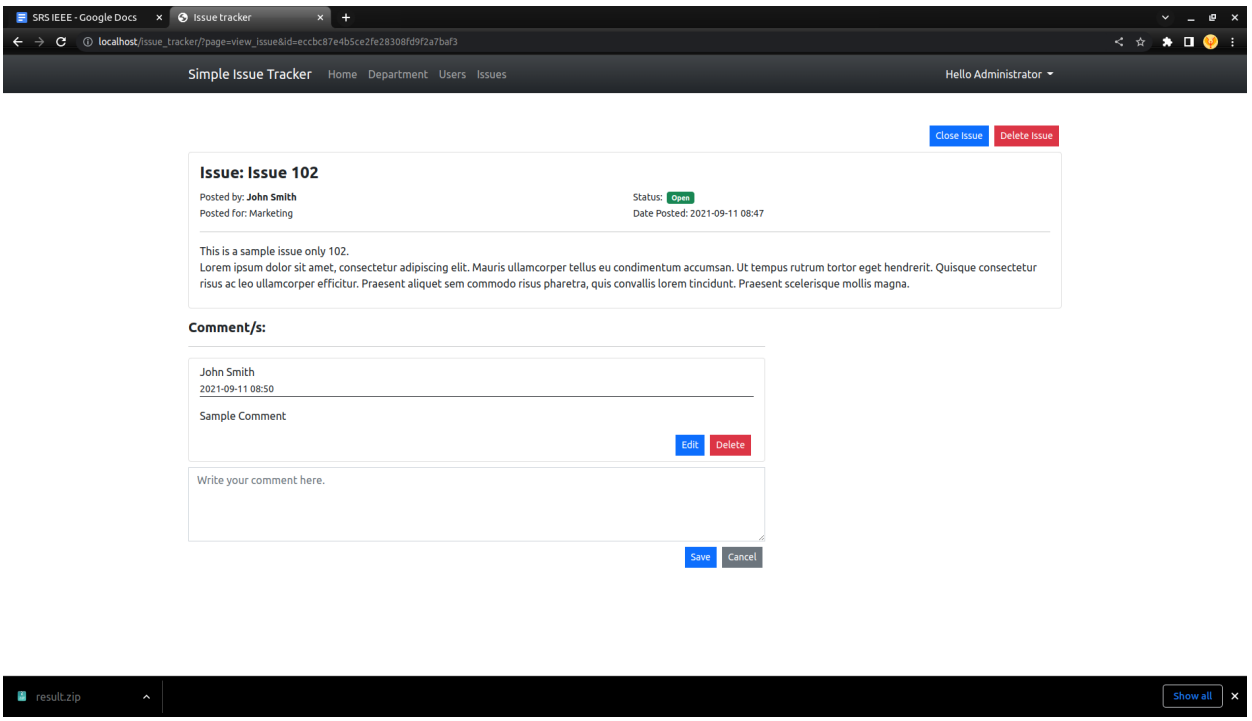
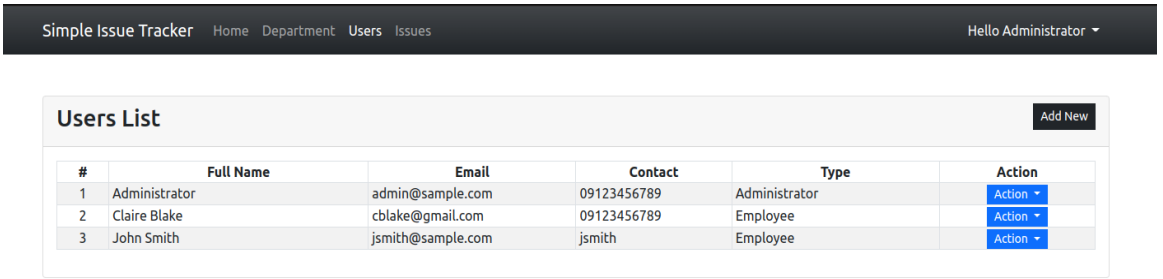
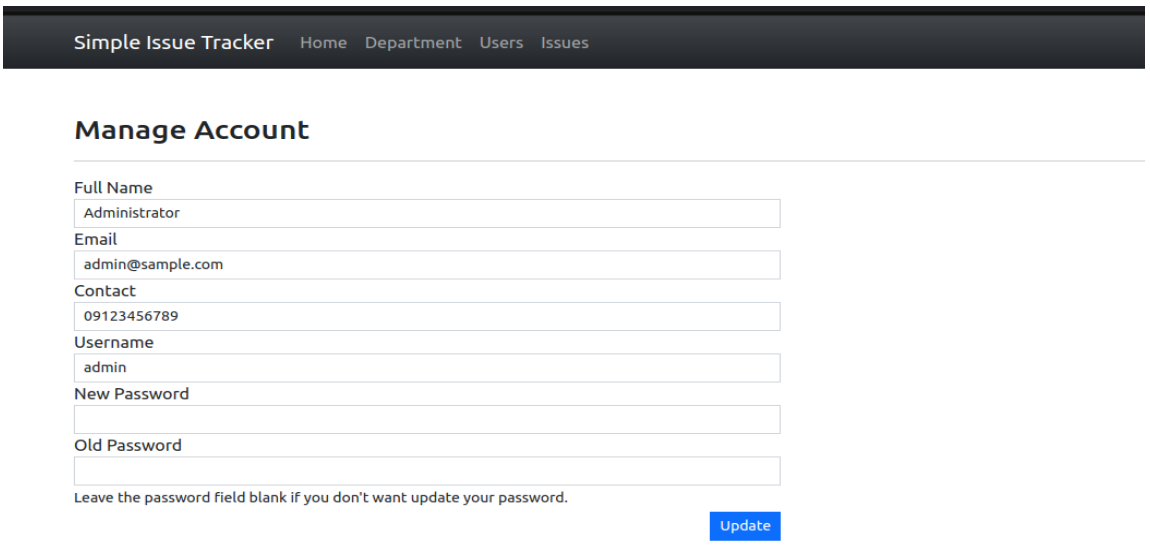


Figure 2: Bug Issue with Description and Comments

Description of the particular issue is shown along with the department name and the name of the employee who releases the issue. An option to close the issue will be provided in case the problem has been solved. A delete issue option is also provided in order to delete an opened issue. Comments have been provided to review the issues and provide interaction between departments. It can also be assigned to a specific user and therefore add more information to the user who has to address the issue.



The users list is a view which allows the administrator to manage the view of the issues addressed in the company. Every action performed by every user is taken into account and documented in the list form so as to let the end product be traceable. The administrator can modify changes onto the user list and make any sort of changes as per requirements.



This view comes in handy when you change the department, leave the job or edit the profile details(eg: changing the passwords, username, contact,etc.)

3.2 Functions

System functional requirements are specified by use cases and specific requirements. The use case helps understand system behavior, and the specific requirements extend the information from the use case.

Use Case: 1 Create a bug issue and save it

CHARACTERISTIC INFORMATION
Goal in Context: The user will be given an interface to upload the details of the bug and save it in the database. This can only be modified by the user who created it. Other users can add comments depending on their thoughts
Scope: Bug Reporting System
Level: Primary task
Preconditions: User has an account & has logged in successfully, or has finished creating an account . User is not the administrator. There are no outstanding invitations that conflict with the exclusion set.

Minimal Guarantee: Any invitations that the user accepted or declined have been returned to the administrator. Others are left outstanding. Any preference/exclusion set changes that the user committed are stored in the system.
Primary Actor: Attendee
Trigger: User logs into system

MAIN SUCCESS SCENARIO

1. System shows all bugs(solved/unsolved), user's current progress, and provides options that allow user to:
    - accept/decline zero or more bugs
    - modify preference/exclusion selective bugs
    - check for newly arrived bug issues
    - logout
2. User views , assess the situation based on bug severity.
3. System shows user's current bug reports, and provides options that allow user to:
    - create a new bug issue
    - modify an existing bug issue
    - delete an existing bug issue

Use Case: 2 Update a Bug Issue

CHARACTERISTIC INFORMATION
Goal in Context: If a bug issue is cleared by the user he can mark it "Closed" or even increase the severity if the bug is dangerous.
Scope: Bug Reporting System
Level: Primary task
Preconditions: Separate User Accounts
Success End Condition: Fixed bugs can be later removed to avoid cluttering.
Primary Actor: User

MAIN SUCCESS SCENARIO

1. Users enters information about the bug report:
    - severity of the bug
    - time it was reported
    - comments for the bug report

System Functional Requirements

Some of the basic functional requirements of the bug tracking system include-

- Recurring errors-
  Some of the most common bugs faced and its solution will be bundled in a database, as a result if the user need not wait until their bugs get resolved and can refer to the solution in no time.

- Budget-
  Resolving bugs within a short turnaround time during a project execution enhances team productivity and avoid the lose that would have occurred due to a defective software

- Notifications-
  Automatic notifications sent to the user's email will notify them as soon as their issue is resolved. It also notifies the members when a new bug is reported and they have to fix it.

- Restrict the bug to be resolved by only one member-
  A bug reported can be picked by only one member, the issue status displayed informs the other members that they need to invest their time on that bug unless forwarded to them. This ensures that all bugs are addressed in a shorter time which increases productivity.

- Security-
  Keep the passwords and data encrypted so that other organizations don't access critical data. Most of the private data will be encrypted to follow our client's privacy.

- Delete Old Records and Data-
  Delete Data of the bugs that are cleared helps reduce a lot of unwanted space and therefore increases performance.

Nonfunctional Requirements

- Cross-platform software - The client can access the software on any OS
- Robust and real-time- A user friendly software where all the bugs are addressed and resolved in a short turnaround time
- Security-
  Keep user details secure
- User management-
  Register new user
  Update user details
  Change password
  Reset password
- Dashboard-
  Displays the issue status on entering the ticket number for the user
  Displays recently added issues
- Maintain a database that stores the-
  Detailed description of the issue
  Who created the issue
  Who is resolving the issue
  Date of reporting
  Date of resolving
  Progress report
  Resolution summary