Software Requirements Specification
For the
CTP Issue Tracker

Document # CMSC447-FA2019-G08-SRS-1-0

Version 1.0

14 October 2019

Table of Contents

0 Revision History

Revision #	Description	Initials	Date
1.0	● Initial Release ● Added section specifying the scope of the Issue Tracking tool ● Added Initial Requirements (Pending final approval from customer) ●	KDC	11 October 2019

1 Scope

This specification establishes the functional, performance, and development requirements for version 1.0 of a web application, supported by a database backend, used to track issues and tickets for Clean Technology Partners.

1.1 Identification

Title: Issue Tracker
Version Number: 1.0

1.2 System Overview

1.2.1 Purpose

The purpose of the Issue Tracker web application is to demonstrate the benefits of moving away from a Spreadsheet based issue tracking system. The system is not intended to be a solution, but a demonstration of how these tools would improve upon the current process. Currently, tickets and issues at Clean Technology Partners are tracked using a Microsoft Excel spreadsheet passed between team members using email. This system will utilize a web application and a SQL database to provide a centralized location for team members to open, monitor, update, and close tickets. The intended users of the system are Power Systems Engineers who are seeking an alternative way to track tickets.

1.2.2 Development History

Development of the system began in September 2019, with version 1.0 of the system scheduled for completion in December 2019. The project is sponsored by the UMBC Department of Computer Science and Electrical Engineering, and the development team consists of senior computer science majors at UMBC. If successful, the project will be utilized by Melbourne based Clean Technology Partners to demonstrate the capability of commercial Issue Tracking tools such as JIRA.

1.2.3 Deployment Locations

The only planned operating site for the software is the main office of Clean Technology Partners in Melbourne, Australia.

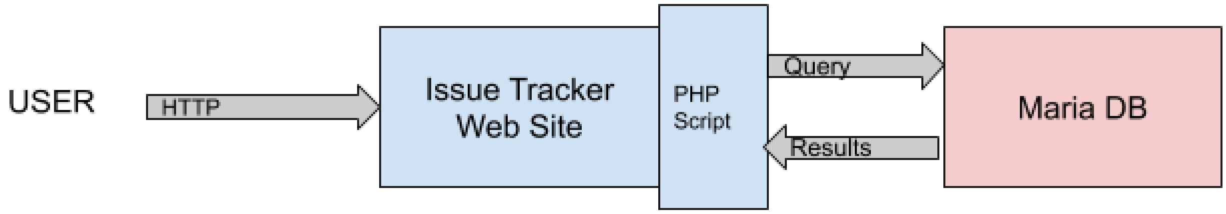
1.2.4 System Functions

The Issue Tracking system includes the following Computer Software Configuration Items (CSCIs)

- Issue Tracker Web Site: The web site for the Issue Tracker system consists of multiple components. A web page which provides the user with a web based front end to view all existing tickets. A web based form allowing users to create a new ticket. A web based form allowing users to modify existing tickets. The web site will be hosted locally on a free web hosting service known as 000Webhost.

- Issue Tracker database: The database for the Issue Tracker will use of MariaDB to store the data for each ticket.

The below diagram depicts the relationship between the Issue Tracker Web Site and the supporting SQL Database.



2 References

The following standards apply:

MIL-STD-498	Military Standard Software Development and Documentation

3 Requirements

3.2 Issue Tracker Functional Requirements

This section describes the functional requirements for the Issue Tracker for Clean Technology Partners' staff (here referred to as "Service Agents"), and the clients Clean Technology Partners serves (referred to here as "Clients"). This section is divided into two parts: a General Requirements section that specifies high-level requirements set by the Clean Technology Partners (referred to here as "The Company"), a Client-Side Web-page Requirements section that describes the user-experience for Clients, a Company-Side Web-page Requirements sections that describes the user-experience for The Company's Staff, and Software Requirements for the Issue Tracker.

3.2.1 General Requirements

3.2.1.1 The Issue Tracker shall be a graphical user interface (GUI) accessed via a website. For testing and demonstrative purposes, the Issue Tracker make use of the web-hosting service 000Webhost at https://www.000webhost.com/.

3.2.1.2 The Issue Tracker shall have an HTML-based system with PHP for the functional application.

3.2.1.3 The Issue Tracker shall allow Clients to communicate software and hardware issues to the Company's Service Agents through a central ticketing system The Client shall be able to create a new ticket from scratch, specify their issue in a Comments Section, and provide the name/contact information of a Service Agent to assist them.

3.2.1.4 The issue tracker shall alert the Service Agent when a Client has submitted a ticket to the Issue Tracker with their contact information.

3.2.1.5 The aforementioned ticket shall be saved in a central database for the Client and Service Agents to view.

3.2.1.6 A ticket shall have a set of statuses it may go through: "Open", "In-Progress", "Closed - Pending Review", and "Closed".

3.2.1.7 The Issue Tracker shall allow the Service Agent to mark the issue as "Closed - Pending Review", and then alert the appropriate Client of the issue's status. Provided that the Client is satisfied with the Service Agent's solution, and that they believe the issue has been resolved, the Issue Tracker shall allow the Client (and only the Client) to close an issue by setting that ticket's status to "Closed"

3.2.2 Client-Side Web-page Requirements

3.2.2.1 The Client shall be provided with a "Main" welcome page that includes an "About this Website", "Help", and "Ticket Manager" section.

3.2.2.2 Clicking the "About this Website" option shall take the user to a page that includes a brief introduction to the Issue Tracker, it's purpose, and the services it offers to the user.

3.2.2.3 Clicking the "Help" option shall direct the user to a page with helpful instructions for creating a ticket, viewing the ticket dashboard, and closing a ticket.

3.2.2.4 The "Ticket Manager" option shall provide the user with a page containing a link to "Create a New Ticket" at the top and a dashboard containing a table-format listing of all the tickets issued by the user.

3.2.2.5 Selecting the "Create a New Ticket" option shall provide the user with a ticket "box" with required fields for the user to fill-in such as: "Issue Title", "Service Contact" (to specify the email/department to look into the issue), and "Comments" (for the user to specify the exact issue). A "Submit" button shall be provided at the bottom of the ticket "box" for the user to click once the information has been filled out. Upon “submitting” the ticket, a date and timestamp shall be attached to the ticket. See Figure 1 below.

New Ticket	
Issue Title:	(*user provides a title for issue here*)
Owner:	(*user provides Service Agent contact info*)
Comments:	(*user provides comments on their issue*)

Figure 1: “Create a New Ticket” box for Client

3.2.2.6 Within the “Ticket Manager” page, the dashboard table containing all the tickets shall be organized according to the following: 1) the ticket number, 2) their ticket’s title, 3) the ticket status, 4) the ticket’s associated Service Agent (provided by the user when they created that ticket), and 5) a link called “Comment History”.

3.2.2.7 The ticket number shall be in chronological order in which the ticket was submitted (for example, the first ticket issues shall be called “1”, the next shall be called “2”, and so on).

3.2.2.8 The ticket title shall be the title name of the ticket issued by the Client (see Section 3.2.2.5).

3.2.2.9 The ticket status will display the current status of that ticket (see Section 3.2.1.6).

3.2.2.10 The Client shall be able to select any ticket’s status and be provided with a window that offers the options to “Close” the ticket. Upon selecting “Close”, the ticket’s status on the dashboard shall be signed as “Closed” and the appropriate Service Agent shall be notified of the closure.

3.2.2.11 The ticket’s associated Service Agent shall be that provided by the Client when the ticket is first created (see Section 3.2.2.5).

3.2.2.12 The ticket’s Comment History shall include a link (titled “Comment History”). Selecting this link shall send the Client to another page that will display, in table-form: 1) the ticket number, 2) the ticket title, 3) the ticket status, 4) and the ticket history. The ticket history shall contain cells for each comment provided by both the Client and the Service Agent. One cell shall contain one comment submission. Each comment shall be ordered chronologically from most recent at the top to least recent at the bottom in order that the comments were submitted. See Figures 2 and 3 below

Ticket Number	Ticket Title	Ticket Owner	Ticket Status	Ticket History
1	(generic title)	John Doe	Open	(URL)
2	(generic title)	Jane Doe	In-Progress	(URL)
3	(generic title)	Hrabowski	Closed - Pending Review	(URL)

Figure 2: Example of Dashboard

(Ticket Title) History	
Date	**Comments**
Nov 14 2019	(comments here)
Oct 29 2019	(comments here)
Oct 27 2019	(comments here)

Figure 3: Example of Comment History Display

3.2.2.13 If the Client receives an alert that the Service Agent has changed a ticket's status to "Closed - Pending Review", the Client shall have the option to set the ticket's status to "Closed" or "In-Progress"

3.2.3 Company-Side Web-page Requirements

3.2.3.1 The Service Agent shall be provided with a "Main" welcome page that includes an "About this Website", "Help", and "Ticket Manager" section.

3.2.3.2 The "About this Website" shall be the same as that described in Section 3.2.2.2.

3.2.3.3 Selecting the "Help" link shall direct the Service Agent to a page that describes the Ticket Manager dashboard, how Clients issue tickets, how to update the status of a ticket, how to suggest a ticket be closed, and will disclose the fact that only the Client may "Close" a ticket to the Service Agent.

3.2.3.4 Selecting the “Ticket Manager” section shall provide the Service Agent with a table-form dashboard similar to that provided to the Client in Section 3.2.2.6. All “Open” tickets (those that remain open at any stage mentioned in Section 3.2.1.6 shall be open to modification by the Service Agent. The Service Agent may select a ticket’s “status” and will be provided with the following options: 1) “In Progress”, and 2) “Closed - Pending Review”. The Service Agent shall not be able to declare a ticket “Closed”. Instead, upon setting a ticket’s status to “Closed - Pending Review”, the Client owner associated with the ticket shall be notified of the status update. The Server Agent shall be forced to wait for the Client to declare the ticket status as “In Progress” or “Closed”. A status return of “In Progress” indicates that the Client is not yet satisfied and requires this ticket to remain open; a status return of “Closed” shall indicate that the ticket issue has been resolved and is no longer an issue.

3.2.3.5 The Service Agent shall also have the ability to read the Comment History of a particular ticket as described in 3.2.2.12.

3.2.4 Issue Tracker Software Requirements

3.2.4.1 The Issue Tracker shall use HTLM web pages as a base for the GUI.

3.2.4.2 The Issue Tracker shall make use of the web-hosting services provided by https://www.000webhost.com/ during the testing stages.

3.2.4.3 The actual “ticket” objects that are the central components of the Issue Tracker shall be PHP objects.

3.3 Issue Tracker external interface requirements

3.3.1 Common Interface

All interactions - for both Clients and Service Agents - shall be with a single, web-based GUI that makes use of a server hosting both the interactive web-pages and their functionalities (main pages, ticket managers, help, etc.), as well as the database storing the ticket data. This platform for server storage of the web-pages and the database shall be over the https://www.000webhost.com/ web-hosting service.

3.3.1.1 Common Interface Server and Database Requirements

As the above-mentioned web-hosting service provides both server space and means for storing a database, there will be no need to supply a server or custom-made database for the Issue Tracker.

3.4 Issue Tracker internal interface requirements

The Issue Tracker shall make use of PHP Data Structures (specifically, PHP objects) for the actual tickets. The PHP ticket objects shall contain variables that contain the information outlined in Section 3.2.2.5 above.

3.5 Issue Tracker internal data requirements

The Issue Tracker shall make use of the database capabilities provided by the web-hosting platform mentioned in Section 3.3.

3.6 Issue Tracker environment requirements

The computer acting as the server and host for the Issue Tracker site shall meet the basic requirements of an HTTP server, as outlined in RFC 2616.

The Issue Tracker website shall be accessed through Google Chrome.

No specific Operating System has been specified for the web-hosting service. The Client and the Service Agent may use any version of Windows or any distribution of Linux that provides a Graphical User Interface to access a web browser.

No CPU processor requirements have been identified.

No RAM requirements have been identified.

3.7 Computer resource requirements

The computer acting as the server and host for the website shall enable the use of HTML and PHP, as well as provide a means for maintaining a database.

3.8 Software Quality Factors

Testability - Failure review testing shall be conducted at regular intervals during the life-span of the project’s code development from start to finish (the regularity will be determined by whether significant functional pieces of the code were added since the last test). The code shall be integrated into the web-hosting platform, where developers will test to ensure the following: 1) that a user may have access to the web-site, 2) that the specific web-page to be tested provides the correct functionalities outlined in Section 3.2, 3) and that the system does not crash at any point during the user's session. Significant functional developments (to be specified during testing) in the codebase of the Issue Tracker shall be added to the web-hosting site for trouble-shooting purposes and demonstrations for the Customer.

Usability - A user-friendly experience is critical for the Customer. To ensure that the Issue Tracker is usable, the Customer shall be provided with demos of the latest version of the web-site throughout the Issue Tracker's development. Customer feedback shall be documented and the developers shall negotiate further developments.

Maintainability - All code for the Issue Tracker shall be properly documented and maintained in a central GitHub account for developer access. The code shall include at least one comment per five lines of code. Each function shall also have comments that include a "Function Description" descriptor that adequately explains the function's purpose and implementation.

3.9 Design and Implementation Constraints

3.9.1 Time and distance between the Customer and the Developers will be a major constraint in this project development. As the time difference between Baltimore, MD and Melbourne, Australia is a 14 hours, meeting times will be constrained to a one-hour interval every Monday night at 8:00pm E.D.T and Tuesday at 11:00am Australian E.D.T. These meetings will occur once a week. Demonstrations and requests for additional features to the Issue Tracker will be limited.

3.9.2 As the Customer does not own a server to host the web-tracker, the Issue Tracker and its associated database will (for the foreseeable future) need to utilize a web-hosting service.

3.9.3 The web-hosting service that will run the Issue Tracker requires Google Chrome as a browser; therefore, any testing and demonstrations will need to be done through Google Chrome. Use of the Issue Tracker by the customer will also have to be through Google Chrome.

4 Qualifications Provisions

Requirement Number	Demonstration	Testing	Special
3.2.1.1	YES		
3.2.1.2	YES		

3.2.1.3	YES		
3.2.1.4		YES	
3.2.1.5		YES	
3.2.1.6	YES		
3.2.1.7	YES		
3.2.2.1	YES		
3.2.2.2	YES		
3.2.2.3	YES		
3.2.2.4	YES		
3.2.2.5	YES		
3.2.2.6	YES		
3.2.2.7	YES		
3.2.2.8	YES		
3.2.2.9	YES		
3.2.2.10	YES		
3.2.2.11	YES		
3.2.2.12	YES		
3.2.2.13	YES		
3.2.3.1	YES		
3.2.3.2	YES		
3.2.3.3	YES		
3.2.3.4	YES		
3.2.3.5	YES		
3.2.4.1	YES		
3.2.4.2		YES	
3.2.4.3	YES		

5 Requirements Traceability

All requirements are provided in section 3. Each has a unique number. All future documentation of the Issue Tracker shall be traceable to the requirements provided in section 3.

6 Notes

6.1 Background and Rationale

The purpose of this software is to provide a Proof-of-Concept (POC) to Clean Technology Partners Pty Ltd. Currently, the staff at Clean Technology Partners track issues, progress, comments, and issue closures through a Microsoft Excel spreadsheet that is emailed among employees. As Clean Technology Partners accepts legal contracts and its staff must be able to efficiently meet the needs of its clients, this approach can only cause confusion among staff and is, therefore, detrimental to the organization's progress. A new, organized, centralized system must be implemented.

The Issue Tracker software offers a simple and effective method of organizing how employees at Clean Technology Partners manage tasks and communicate with customers. With an easy means of creating a ticket, a Client will have an easier time alerting the appropriate staff member of their needs. And, with a dashboard system to visually display active tickets, who issued them, and the progress made with those tickets, the staff at Clean Technology Partners will be able to address the needs of their clients more reliably and efficiently.

As Clean Technology Partners does not appear to have a server on hand to host the Issue Tracker, and requires that the Issue Tracker be intuitive and easy-to-use, the developers on this project have opted to make use of a publicly-available, easy-to-set-up account on 000Webhost. The developers of this project also reasoned that given the POC nature of this project, it would be best to use HTML-based web-pages that offer basic text and URL linking capabilities, as well as PHP for making simple objects to create the actual tickets that are central to the Issue Tracker.