STATUS TRACKING SYSTEM

Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to present a detailed description of the Status Tracking System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

1.2 Scope

Status Tracking System helps to assign and track the status of the incident tickets assigned to a resource. Incident is an unplanned event that causes interruption in service or deterioration in service quality. It maintains the history of tickets assigned to a resource and helps users to prioritize the tickets assigned to them. Also, the system consists of many useful functions and features which make handling of tickets easy.

This system increases productivity by enabling better co-ordination between teams. It helps on minimizing time wasted on tracking issues. Also, improves quality by ensuring all tickets are recorded and handles the end to end progress. Gives users the freedom to access from mobiles or PC.

1.3 Glossary

Term	**Definition**
Ticket	An issue to be resolved.
Resource	An individual who handles tickets.
Database	Collection of all the information monitored by this system.
Admin	An individual who handles the entire application.
Developer	An individual who resolves a ticket.
Tester	An individual who tests a ticket.
Report	Displays the overall progress of the ticket.
Priority	Shows the request in which the ticket needs to dealt with.
Status	Shows the present state in which the ticket is.
Remedy	It’s a state which progress to do documentation else close the ticket.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Status Tracking System	Helps to assign and track the status of the incident tickets assigned to a resource.
Delete	An activity performed to remove the ticket.

Table 1.1: glossary

1.4 Reference

[1] https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html

[2] https://docs.gradle.org/current/userguide/maven_plugin.html

[3] http://javabeginnerstutorial.com/hibernate/hibernate-framework-basic/

[4] https://datatables.net/development/server-side/jsp

[5] http://www.javatpoint.com/example-to-connect-to-the-mysql-database

[6]http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/basic_app_embedded_tomcat/basic_app-tomcat-embedded.html

[7] https://mvnrepository.com/artifact/org.springframework/spring-webmvc

[8] http://viralpatel.net/blogs/spring-4-mvc-tutorial-maven-example/

[9] http://projects.spring.io/spring-webflow/

[10] http://www.oracle.com/technetwork/database/mysql/index.html

[11] http://www.oracle.com/technetwork/java/javaee/overview/index.html

2. OVERALL DESCRIPTION

2.1 Product Perspective

Status Tracking System stores the following information as shown below.

- Ticket Details:
  It includes ticket id, ticket description and application name of a ticket. Further we can modify each ticket by setting priorities, status and giving comments. The admin handles the tickets and can allocate the tickets to a resource.
- Resource Details:
  Here resources are Admin, Developer and Tester who handle the tickets. They are assigned with certain work and managed accordingly. The completed work is present at closed tickets.

2.2 Product Functions

The three actors in the product are Admin, Developer and Tester. The main objective of STS is to raise tickets by the Admin and assign it to the developers. Each ticket has priority to work on it, type of the application and description. Once the ticket is assigned to the developer, they will work on it. Each ticket comprises of various phases such as analysis, build, deployment, design, documentation, estimation, testing and UAT. Once the activity of the ticket reaches the estimation

phase, the ticket must be proceeded to tester with the activity of testing. The tester will complete the work on it and the activity is changed as UAT which reflects the work as completed from the tester side. Again, the developer looks on the same ticket and sets the status as closed. There are various statuses available such as in-progress, on-hold, approved, completed, closed and not started. Thereby if the status of the ticket reaches closed state it goes to the documentation part. Once the status of the ticket is being updated on the REMEDY tool then we can update the status of remedy as closed, by which the ticket life cycle is completed. Else the remedy status will be in progress which means there is still something which needs to be looked upon regarding the ticket.

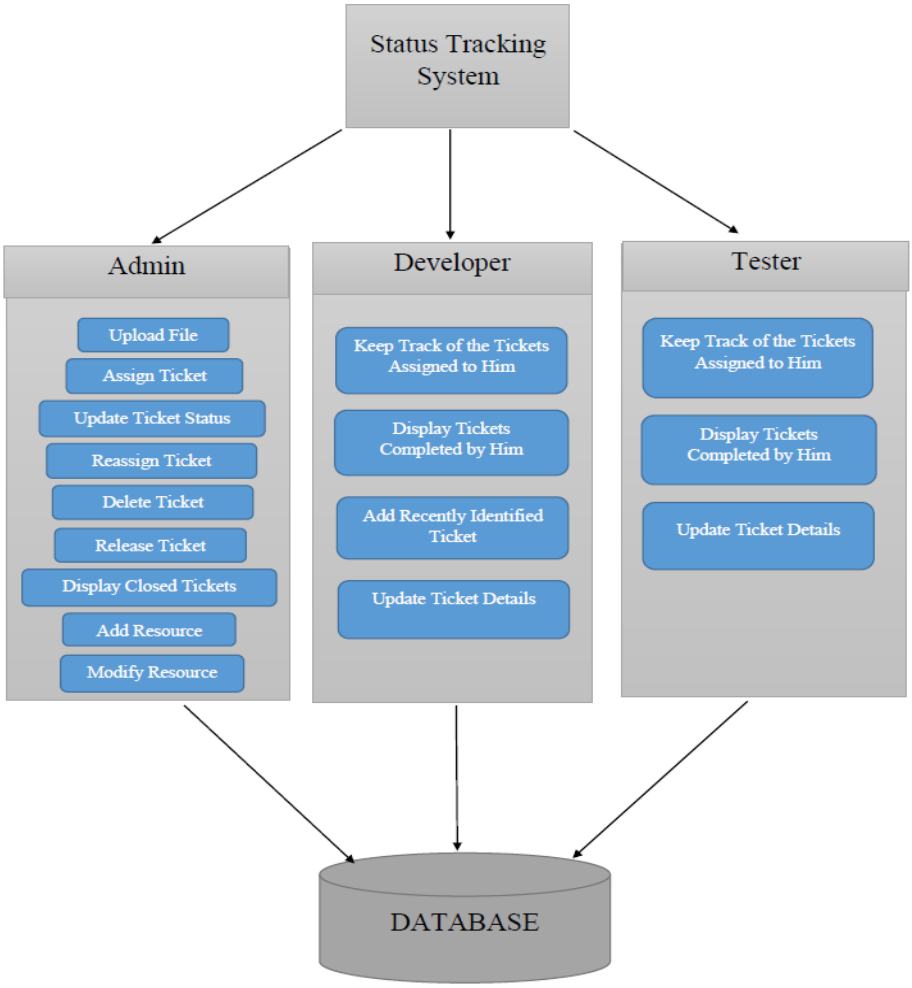

Figure 2.1: Context Diagram

2.3 User Characteristics

Admin: Can Add/Modify resource in the system and has rights to set admin and role preferences to a resource.

Developer: Can view his assigned tickets and set status. Once the estimation phase is done he can send the task to the tester.

Tester: Can view the tickets which is being assigned by the developer. After resolving the ticket, he can set the status UAT which is final.

3. REQUIREMENT SPECIFICATION

3.1 User Interfaces

The interface of this system is flexible and reliable. Navigating through the various pages is easier. The various tasks like getting reports, adding resources and deleting tickets is possible with just a click.

Figure 3.1: Login Page

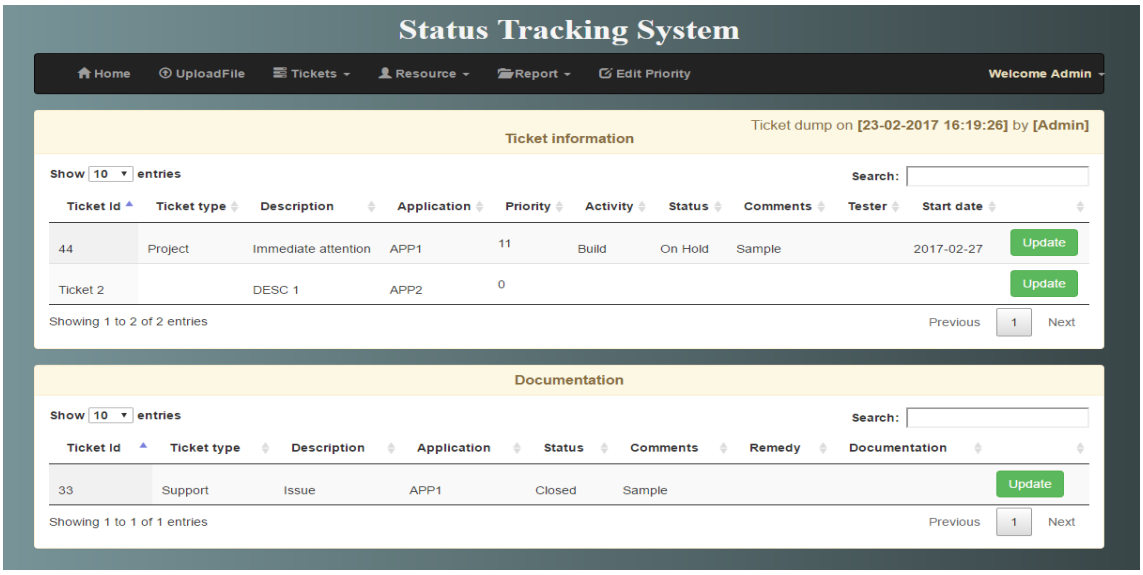

Figure 3.2: Admin's Home Page PC View

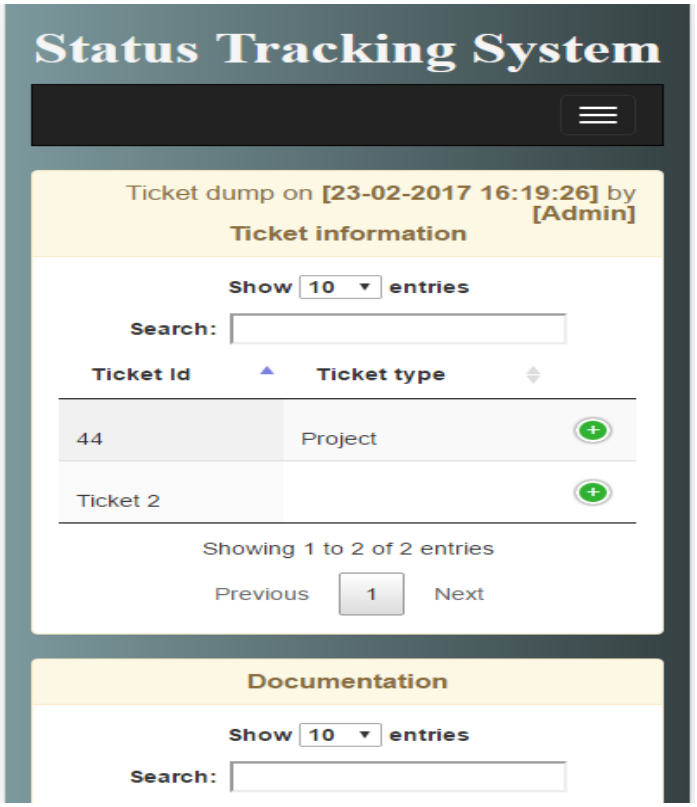

Figure 3.3: Admin's Home Page Mobile View

3.2 Hardware Interfaces

The physical system is installed with the application and its then connected to the database. This creates a network for the usage of the system.

3.3 Software Interfaces

The application is being built in the Eclipse IDE where the IDE is installed on the Windows OS which acts as a platform for the Eclipse IDE and the Apache Tomcat server. The data of STS is maintained in MySQL database.

4. FUNCTIONAL REQUIREMENTS

4.1 Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The actors usually individuals involved with the system defined as per their roles.

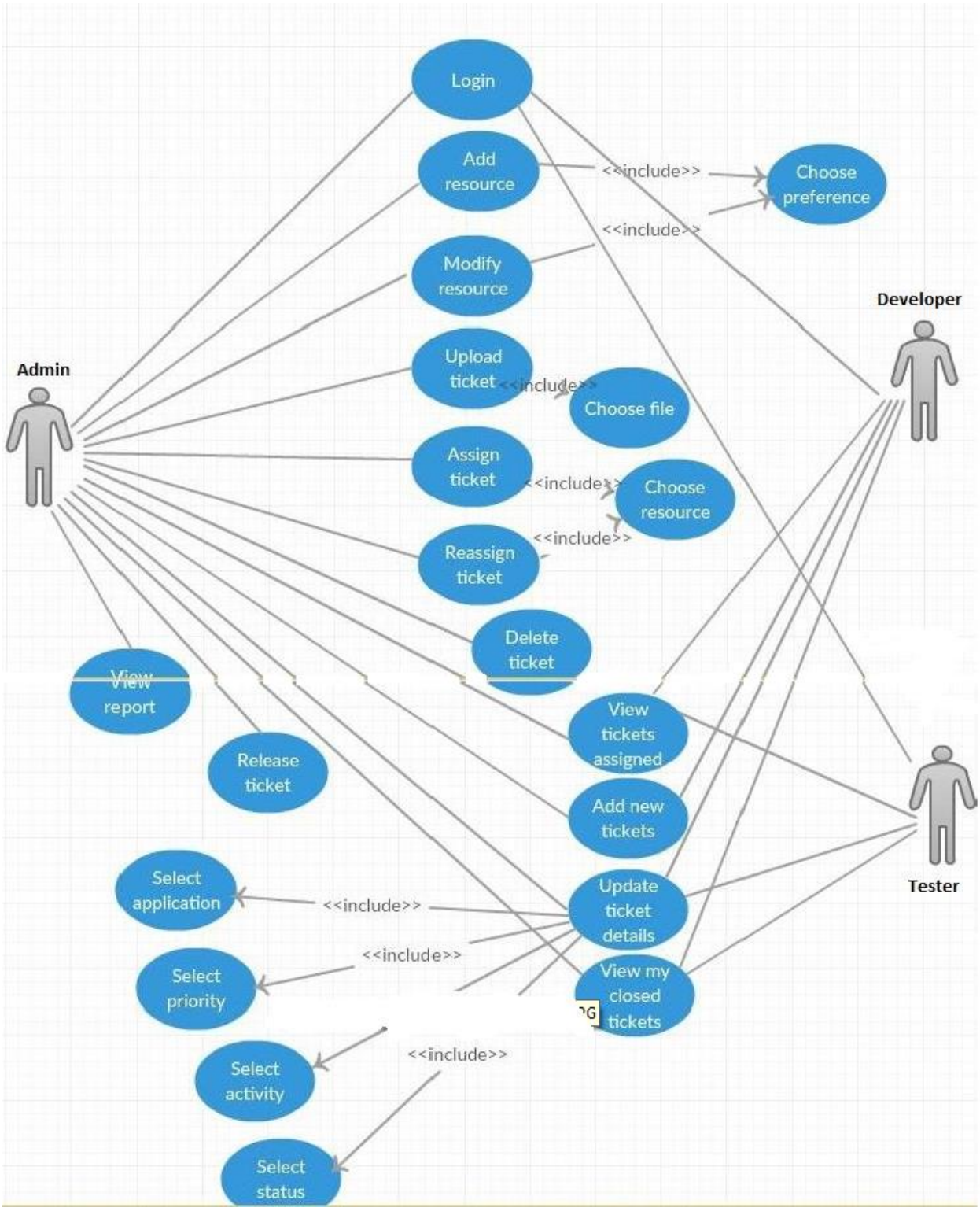

Figure 4.1: Use Case Diagram

Description

The above use case diagram depicts the functionality of the actors in the system. The actors in this system are admin, developer and tester.

Admin

The admin holds the major role in this application wherein he has the right to add / modify details of the resource, upload tickets, assign/reassign tickets, delete tickets, release tickets and view various reports on resources and tickets.

The admin can add a new resource and set certain preferences for the resource such as admin preference and tester/developer preference. Later the admin can also modify these actions if necessary.

The admin can upload tickets to the system which will be display on the assign ticket page. From here certain tickets are reassigned to various resources which are available accordingly.

If there are some tickets which are redundant then the admin has the right to delete those tickets. Also, the admin can release certain tickets to different resources and view the reports of what is happening with each ticket from the report pages.

Developer

The developer holds some functionalities such as keeping track of all the tickets assigned to the developer, displaying all the closed tickets, adding recently identified tickets and updating ticket details.

Once the developer has a ticket assigned then the developer will be working on it and will set statuses accordingly. After certain period when the status of the ticket reaches closed then that ticket is completed and it will be displayed on my closed tickets page. If suppose the developer finds any new small issues in the application, they can add their own ticket and keep a track of that also.

Tester

The tester also holds some functionalities such as keeping track of all the tickets assigned to him, displaying all the tickets completed by him and updating ticket details.

The tester gets the tickets which are assigned by the developers. Once the tester has finished the testing process they change the status to UAT which means the whole process with that ticket is completed. The tester can view the tickets which are completed and can also update the ticket details.

4.2 DFD diagram

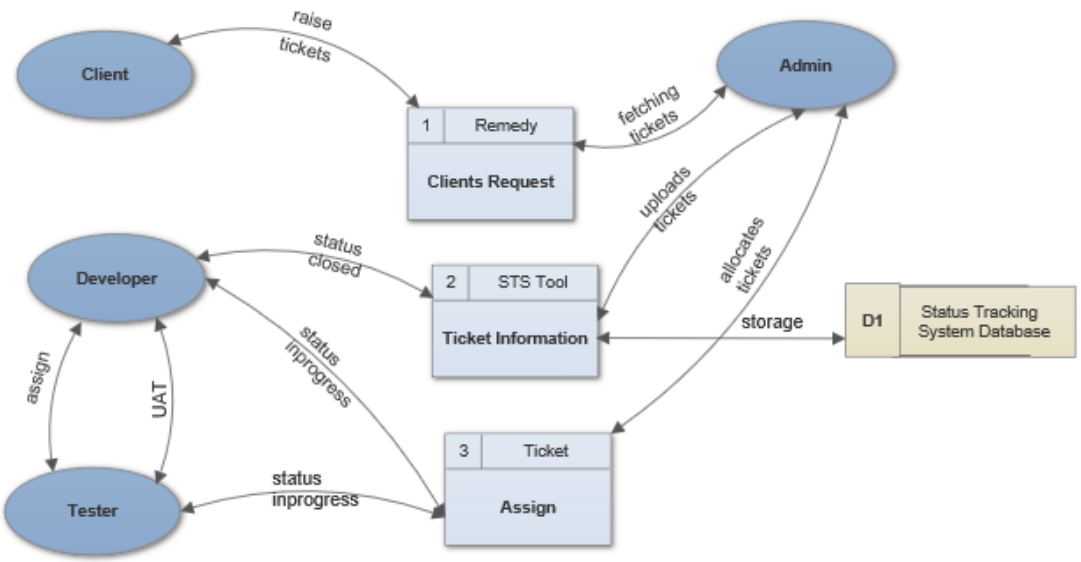

Figure 4.2: Data Flow Diagram

A data flow diagram is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of processing. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

The flow starts with the login page where developer/tester/admin login with their credentials. If the credentials match with the database, then the control goes to access STS application.

4.3 Functional Requirements

The system shall allow administrator to add and modify resource, upload ticket file to the system, assign/reassign tickets to the resources, delete and release tickets. The system shall allow administrator to view ticket status report. The system shall allow administrator to view status report on the day to day basis.

The system shall allow administrator to view summary of the report by providing any one of the following data like employee name, application, activity, status, release, start date, end date, ticket

type and tester name. The system shall allow administrator to view ticket status by choosing either ticket id or resource name and allow him/her to edit the priority of the ticket.

The system shall allow administrator to view ticket for the day, ticket history status, clarification, employees who have not reported and tickets with no status change. The system shall allow developer to view all the tickets assigned to him/her and all his/her closed tickets. The system shall allow developer to add recent tickets, update ticket details and ticket status.

The system shall allow tester view all the tickets assigned to him/her and all his/her closed tickets, update ticket details and ticket status. Priority of the tickets assigned to the resource should be unique.

5. NON-FUNCTIONAL REQUIREMENTS

Previously the status of a ticket would be reported on an Excel sheet which was a cumbersome job for the higher officials in the project to track down. They were unable to monitor the resource progress and there used to be a lot tickets with unchanged status. Also, there were storage constraints. As STS came into picture all these issues were resolved and the ticket management became easier and confined. Managers could track down the progress on tickets and resources on a day to day basis by getting a report. There is a lot of flexibility.

5.1 Usability

The system shall allow the users to access the system from a browser using HTML or it's derivative technologies. Since all users are familiar with the general usage of browsers, no specific training is required. The system is user friendly and self-explanatory. Also, the mobile version of this application has made it more flexible for the users.

5.2 Reliability

The system must be very reliable due to the importance of data and the damages incorrect or incomplete data can do.

5.3 Availability

The system is available 100% for the user and is used 24 hours a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Mean Time Between Failures (MTBF)
The system is developed in such a way that it ***may*** fail once in a year.

Mean Time to Repair (MTTR)
Even if the system fails, the system will be recovered back up within an hour or less.

Accuracy
The accuracy of the system is limited by the accuracy of the speed at which the resources of the system use the system.

5.4 Performance

Response Time
The Information page should be able to be downloaded within a minute. The system shall be allowed to take more time when doing large processing jobs.

Administrator
The system shall take as less time as possible to provide service to the administrator.

Capacity
The system can handle many users.

Resource Utilization
The resources are modified according the user requirements and as per the books requested by the users.

5.5 Regulatory/Compliance Requirements

The system should limit access to unauthorized users. The system should allow the administrator to delete ticket comments from the database if required.

5.6 Security Requirements

Members other than the administrator can view the tickets assigned to themselves and add recent tickets but should not be able to delete any of the tickets. Tester should not be allowed to add or delete tickets. Developer and tester should not be allowed to add resource on their own and to view tickets that are not assigned to them.

6. PRIORITIZATION

In Status Tracking System, prioritizing methods establish the relative value of choices or alternatives. we can prioritize our results in a ranking of the choices to show what should be done first, what requires the greatest attention, and what needs the most resources. Methods differ

depending on whether the priorities are based on objectives or criteria. In STS, following are the priorities in which issue can be resolved based on their values.

Priority-1: Ticket will stop STS progress if not resolved. The impact of this issue will lead to potential project stoppage.
Example: Issues raised by board members about the financial viability of the STS project are preventing the project from moving forward as planned.

Priority-2: Ticket will likely move the STS project back in terms of budget or timeline, or will materially affect quality or scope. The impact of this issue leads to not meeting the client's requirements and possibility of project work not completed on time.
Example: The STS project is short on a specific skill set.

Priority-3: Ticket will have material effect on STS, has potential to be moved to high category and/or requires significant resources to manage. The impact of this issue leads to possibility of project work not completed on time.
Example: Negotiations with functional managers in an organization competing for scarce human resources are forecasted to delay STS project completion.

Priority-4 and others: Issue is expected to have a moderate effect on the STS, but will require resources to address.

7. RELEASE PLAN

New patch releases of STS are released most frequently. They are released as required, depending on how important the changes are, and if it fixes security issues. New minor releases of STS are released as per demand, how important the changes are, and if it fixes security issues.

The aim is to release these, not more than one a month. A best effort will be made to keep up with the latest version of the SRS always, within the requirements specified here.

Any small changes or bug reports, that don't require changes to these specifications, should be submitted to the GitHub issue tracker. Any STS should consider if it requires changes to these specifications, and the release requirements mentioned in this section. The process of creating a STS involves community consultation.