



IT-Asset Service Tracker

SRS Document

IT Asset Service Tracking	3
Introduction	3
Management Summary	3
Key Assumptions	4
High Level Architecture	4
Functional Requirements	5
Use Case Diagrams	5
Use Cases	6
<i>Maintain Masters</i>	6
<i>Logs a Service Request</i>	7
<i>Update Service Request</i>	8
<i>View Service Dashboard</i>	8
<i>Update Contract Status</i>	9
Escalation	10
Preventive Maintenance Schedule	10
Update Preventive Maintenance Tickets	10
Logical Object Model	11
Database Design Guidelines	12
Testing Approach	12
Suggested Technical Reading	13

Disclaimer

This Software Requirements Specification document is a guideline. The document details all the high level requirements. The document should be used as a guideline by the students to design the Solution Architecture for the project. The document also describes the broad scope of the project and high level logical object model. But while developing the solution if the developer has a valid point to add more details being within the scope specified then it can be accommodated after consultation.

IT Asset Service Tracker

Introduction

The purpose of this document is to define scope and requirements an Information Technology Helpline system that lets the users report faults for the IT assets issued to them. The application will track the IT assets maintenance contract renewals, Preventive maintenance schedules and Service requests management for the IT Asset Service Tracking of a In-house team of engineers.

This document should be used by the development team to architect the solution the project.

The project provides an exposure to challenges of having a large IT infrastructure with various levels of users. The kind of support that is expected from the support staff. The performance of the support staff is directly related to how much of loss the business experiences when the users are not able to get their work done on time due to service issues.

Management Summary

An infrastructure organization is expanding its operations and soon they will be having over 3000 employees in their Delhi office. Almost all the employees are allotted laptops or desktops, there are shared printers, some employees have USB sticks for internet access while traveling or working from home. Though the organization promotes less paper generation, still some departments have their own dedicated printers for privacy of content to be published. The users have an intranet for their self service apps and as well as business applications that allow them to focus on the core work area.

Even though it may sound simple, but tracking of IT assets is one big activity, looking into maintenance renewals, bulk purchases etc are quite challenging to track on daily basis.

To ensure business continuity and quick issue resolution, the management decided to have an automated solution for the following:

1. Users can log in service issues online.
2. The system categorizes issues raised by users, assign SLAs for resolution.
3. The system automatically assigns the service call to the engineer having skills to handle types of issues.
4. Track assets status, their maintenance renewals etc.
5. Monitor SLAs
6. Send timely notifications to Procurement and Service team before the service or renewal milestone are achieved.

The proposed solution will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

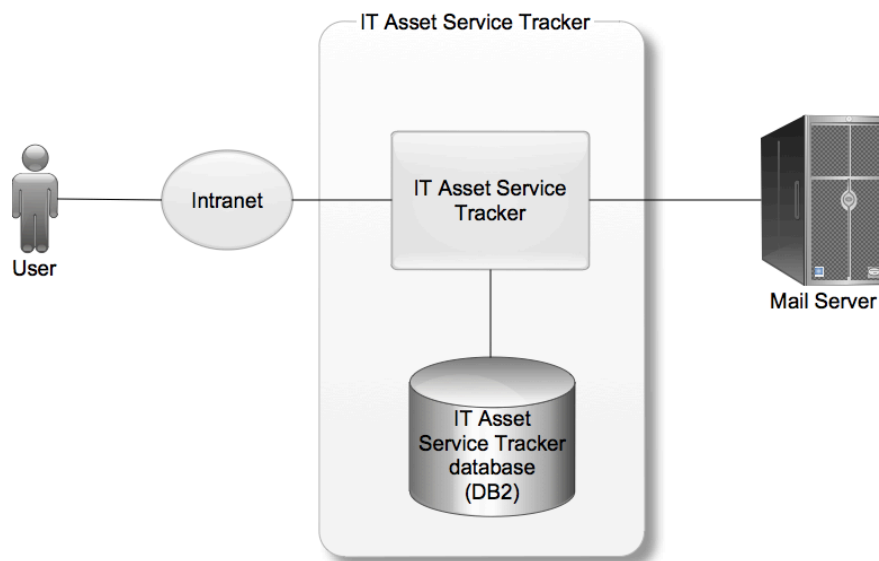
Key Assumptions

1. The call servicing along with asset tracking is a vast system, to keep the scope of development short, only some part of service call handling and related process is being considered. For the asset, in the current SRS, only focus is on the maintenance.
2. The purpose here is to make the developer understand the concept of call allocation, what are the factors that drive the resolution SLA. IT-Asset Services

High Level Architecture

IT-Asset Services high level architecture is illustrated through the context diagram shown below. It will have following categories of users:

1. Service Engineer
2. Users of the company
3. Procurement Manager



IT-Asset Services Context Diagram

IT-Asset Services	System that tracks IT Assets maintenance contracts, Service Calls logging and assignment,
IT-Asset Services Database	Records the Calls, list of engineers, skills, SLA, Wiki entries
Mail Server	Notifications from the system shall be sent by this mail server

Functional Requirements

The high level functional requirements for the IT-Asset Services are outlined in the Use Case diagram described in this section.

IT-Asset Services will provide a secure user-id/password based secured login mechanism to access its services. The details of this are not outlined here. The development team is expected to create these keeping in mind the general practices followed by the web applications. Login will be a prerequisite to use IT-Asset Services. Internal users will be provided user id/password pair separately.

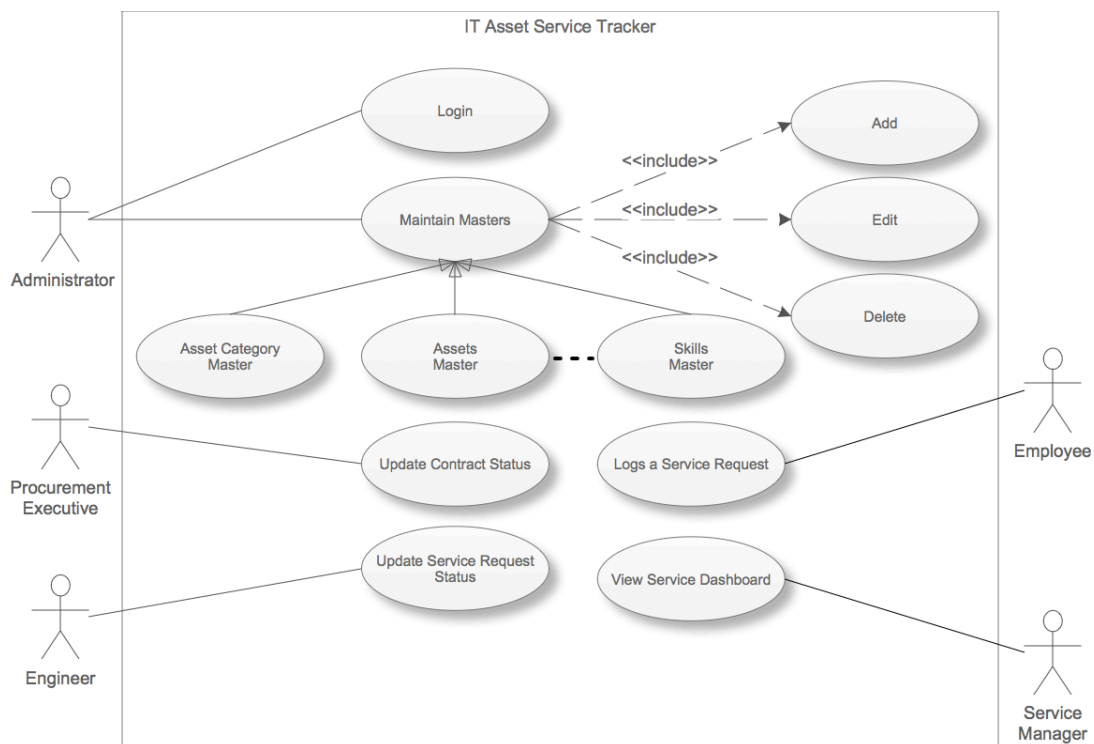
The system will display a different set of menu options for various types of users. The regular employee who is an IT infrastructure user logging into the system shall see, the Status of his/her call if logged already, status of Asset requisition if any new ones have been raised. The feature to log calls and make new requisition shall be available.

The Service team's page will display the queue of allocated calls with their status as per defined workflow. The coordinator should be able to do reassignment of calls. Read escalation mails if any, view the list of Service Engineers reported to duty today.

The Service Engineer should be able to view the calls allocated to him/her through Priority and Normal Queue as separate menu items. List of escalations if any, Reports on Calls performance

Use Case Diagrams

The following figure illustrates the Use Case diagram for the system. The MiS use cases are not detailed here. If the time permits, you may like to generate MIS like Calls with SLA Not Met, Maximum complaints categories, Engineers Calls Serviced.



Use Case Diagram

Use Cases

Maintain Masters

Use Case Element	Description
Number	UC.01
Application	<p>Master maintenance in terms of basic operations viz. add, edit, delete and view. All master maintenance i.e. Asset Category, List of Assets, Employee, Service Engineers, Skills, Engineer skills are child use cases of this Use Case.</p> <ul style="list-style-type: none"> Asset Category: Category id, Category, SLA (in days) List of Assets: Category id, Asset id, Configuration, Date of Purchase, Issued to, Contract Expiry Date, Preventive Maintenance Due Employee master : Employee Id, Employee Name, Date of Joining, Status (Active/In-active) Service Engineers: SE-ID, Engineer Name Skills: Skills id, Skill, Category id Engineer Skills: SE-ID, Skill Id
Use Case Name	Maintain Masters
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the Maintain Masters menu item on the admin interface page
Basic Flow	<ul style="list-style-type: none"> System presents a list of masters that can be maintained. Administrator selects the desired master. System displays a list view and links for <i>add</i>, <i>edit</i> and <i>delete</i>. <ol style="list-style-type: none"> In case add, a new master record data entry form is presented. The master record is saved on clicking the save button provided form clears all the data validations (if any). The list view is updated accordingly. In case of edit, from the list view user is prompted to select the desired record to edit, Selected record is opened for editing. The edited master record is updated on clicking the update button, provided form clears all the data validations (if any). In case of delete, from the list view user selects the check box(s) against each record. Selected records are deleted up on clicking the delete button. However, user is presented a confirmation dialog before deleting the records.
Alternate Flow	<ul style="list-style-type: none"> In event of any error, it is clearly displayed and user is asked to reenter data or perform operation again.
Output	System displays the details of the successful operation.

Logs a Service Request

Use Case Element	Description
Number	UC.02
Application	The call is logged into the system by the user
Use Case Name	Employee logs a service request
Primary Actor	Employee
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the Log a Service Request menu item on the IT-Asset Services landing page
Basic Flow	<ul style="list-style-type: none"> System displays the employee name, current date and time in the header of the form. The second section displays the list of assets i.e. category and asset configuration registered in the name of the employee. A check box is displayed before the asset User Clicks on a checkbox, the system displays a text box to describe the problem. Severity drop down is displayed for the user to select from a pre-defined list (High, Medium, Low, Nice to have) User clicks on Submit button to log the Service request Or Cancel button to abandon the operation. On submit, the system will run a procedure to accomplish following: <ul style="list-style-type: none"> Save the service request with the SR Id (running sequence), employee id, Category id, Asset id, Problem Description, SLA [Will come from SLA as defined in Category Master], Status as 'Unassigned'. Lookup Service Engineer with Skills for the Asset id mentioned in Service Request. From the list of Service Engineers who can service the request, find the engineer with least number of calls in WIP or assigned state and Assign the Service Request to this engineer. The service request for the engineer will get re-prioritized based on the SLA of requests in the queue. The least SLA and priority as 'High' moves up on the priority list. <p>Note: For simplicity purposes only one asset complaint is allowed in single service request.</p>
Alternate Flow	<ul style="list-style-type: none"> In event of any error, the error message displayed and user is asked to reenter data or perform operation again.
Output	System displays the Call Resolution SLA for the asset category selected in the Service Request.

Update Service Request

Use Case Element	Description
Number	UC.03
Application	The Service Engineer views and updates the Service Requests assigned to them
Use Case Name	Update Service Request Status
Primary Actor	Service Engineer
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the Update Service Request menu item on the IT-Asset Services landing page for Service Engineers
Basic Flow	<ul style="list-style-type: none"> • System displays Service requests assigned for the day. • The SR entry displays the Employee name, Asset, SLA, Time Elapsed, Expected Resolution time, Status. • Mouse hover on the Asset will display the problem description. • Status is a drop down field with entries as 'Assigned, WIP, Closed). User can change the status from Assigned to WIP, WIP to Closed only. • The entries where SLA has burst i.e SLA has crossed the Resolution time as t for a category will be displayed with a Red Flag.
Alternate Flow	<ul style="list-style-type: none"> • In event of any error, the error message displayed and user is asked to reenter data or perform operation again.
Output	Email notification is sent to user for the Service Request Closure when status is marked as 'Closed'

View Service Dashboard

Use Case Element	Description
Number	UC.04
Application	The Service Manager views the overall Service requests status and team performance
Use Case Name	View Service Requests
Primary Actor	Service Manager
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the View Service Dashboard menu item on the IT-Asset Services landing page for Service Managers

Use Case Element	Description
Basic Flow	<ul style="list-style-type: none"> • System displays a dashboard kind of display with various sections giving information as follows • Section Current Month: Name of the month say 'July 2012' <ul style="list-style-type: none"> • Total Requests : <Count of requests received> • Resolved within SLA : <Count of SRs closed before SLA time> • SLA Burst : <Count of SRs where SLA was not met> • Section Today: <Today's date> <ul style="list-style-type: none"> • Total Requests : <Count of Request Received> • Assigned: <Count of requests in the assigned state> • WIP :< Count of Requests that are being attended to> • Closed : <Count of Requests that have been closed today> • Section Escalations <ul style="list-style-type: none"> • List of Escalations displaying Employee Name, Service Engineer Name, Complaint. • Section Preventive Maintenance Today <ul style="list-style-type: none"> • Total PM Tickets: <Count of Preventive Maintenance Tickets> • Assigned: <Count of Tickets in the assigned state> • WIP :< Count of Tickets that are being attended to> • Closed : <Count of Tickets that have been closed today> • Section Maintenance Contracts <ul style="list-style-type: none"> • Total Assets Covered : Count of Assets in Maintenance Contract • Contract Expiring Next month : Count where contract is due for renewal next month
Alternate Flow	<ul style="list-style-type: none"> • In event of any error, the error message displayed and user is asked to reenter data or perform operation again.
Output	None

Update Contract Status

Use Case Element	Description
Number	UC.05

Use Case Element	Description
Application	The Procurement user views the assets due for renewal of maintenance contract and updates the contract validity on the IT-Asset Services System
Use Case Name	Update Contract Status
Primary Actor	Procurement Executive
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the Update Contract Status menu item on the IT-Asset Services landing page for Procurement Users
Basic Flow	<ul style="list-style-type: none"> The system displays the 3 tabs of information on the Assets <ul style="list-style-type: none"> List of Assets not under Contract List of Assets renewal due within next month User Wise - Asset List The information displayed in the above tabs will of columnar manner giving Asset Category, Asset id, Contract Expiry Date. The contract expiry date is editable with a date picker. User clicks on Update button to save changes or clicks on Cancel button to abandon operation. The update action will update the contract expiry date column for the record that has been changed.
Alternate Flow	<ul style="list-style-type: none"> In event of any error, the error message is displayed and user is asked to reenter data or perform operation again.
Output	None

Escalation

This is a mechanism for the users to report out any exceptions found in the processing of Service Requests. Normally escalations happen on workflow SLA not met automatically via the system. For this project, create an option to enter the escalation as a complaint logged by the user with reference to a service request. There could be a button on Service request which pops up a text box for the user to enter the complaint. The escalation goes as an email to the Service Manager and also appears in the dashboard for the Service Manager's landing page.

Preventive Maintenance Schedule

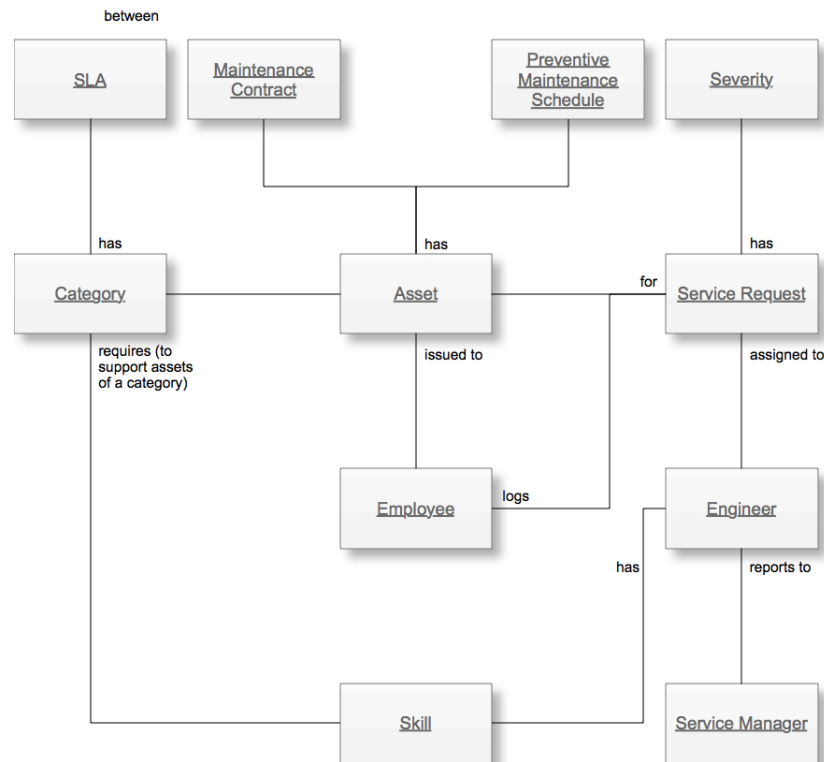
A background procedure to be run every weekend by the Service Manager from the application menu. This process will pull out Assets under maintenance contract and read the preventive maintenance date, if the date is falling in the next week, the the a record is generated for the Asset id along with the date. At the end of this process there is a list of PM due for the next week. The System assigns the Preventive Maintenance tickets to the engineers based on their skill on similar lines as Service Request Allocation.

Update Preventive Maintenance Tickets

An interface to view and update the status of the maintenance tickets on similar lines as the Service Request is required. Use case is not mentioned for the same. The Service Request use case should act as a guideline for the developer.

Logical Object Model

A high level logical object model of the system is shown below. During technical design it will be transformed into a physical model covering all system entities. Such a diagram will include their relationship and its cardinality.



Logical Object Model

1. An Asset belongs to a category called as Asset Category.
2. Each Asset category has a an Service Level Agreement(SLA) for response and fixing of the problem.
3. The employee is issued company assets (IT) that are maintained in the system.
4. IT Assets have a maintenance contract with a validity period. The renewal update is entered in the system
5. A set of Skills are required to trouble shoot or fix a category of Assets
6. A Service Engineer has one or more skills for Asset Category.
7. A Service Engineer reports into a Service Manager
8. All Escalations are viewed by the Service Manager.
9. The Employee raises a service request for an Asset issued to them.
10. A service request has an id, date, Severity as defined by the employee
11. Service request is auto assigned to the Service Engineer based on Skills.
12. The preventive maintenance Service requests are auto generated by the system based on Preventive Maintenance due date.

Database Design Guidelines

This involves the transformation of the use cases, state diagrams, and logical object model into detailed and optimized physical database table designs.

Typically persistent classes will map to table(s) with their attributes as columns of the table. In some cases a high level object may map in to a master-child table. Invoice is one such example where it maps in to "invoice_header" and "invoice_line_item" table.

Associations between two persistent objects are realized as foreign keys to the associated objects. A foreign key is a column in one table that contains the primary key value of the associated object.

Similarly, a standard technique in relational modeling is to use an intersection entity to represent many-to-many associations. Following is a broad checklist for physical database database design:

1. Database must be properly normalized except those instances where de-normalization help improves performance. This option must be used with special care.
2. All persistent classes that use the database for persistency must map to database structures.
3. Many-to-many relationships must have an intersecting table.
4. Primary keys should be defined for each table, unless there is a performance reason not to define a primary key.
5. Indexes should be defined to optimize access.
6. Data and referential integrity constraints should be defined.

Testing Approach

Quality of the software can be achieved with basic hygiene and consistency followed during design and development of User Interface(UI), Navigation, Validations as per the business process requirement.

To ensure the project delivers acceptable quality to the customer, its important to create a checklist of the conventions to be followed across. Common checks as below are for your reference during design and development:

Common Checks	Validation Type
Page Title is valid for the feature being provided on the page	UI
Order of the Data Entry Fields is logical as per the functionality being provided by the feature	UI
Order of the Display only Fields makes viewing and understanding easy for the user	UI
Spellings and Correctness of Label for the Data Entry and Display fields	UI
The labels are not wrapping onto another row thereby adding a blank row on the page	UI
The fields with drop down are displayed in single row instead of drop down coming on the next row	UI
Data Entry field basic validations are working i.e Text field /Numbers / Dates allow data for their type only	Functional
The dates are following a standard format dd/mm/yy on all forms	UI
The color scheme of all forms i.e headers labels , alerts, entry fields are uniform throughout the application	UI
The action buttons for a New Data Entry Form are uniform for all forms that is allowing data entry	UI

Common Checks	Validation Type
The action buttons are performing the desired action e.g. "submit" is creating a new record if there are no errors and recording all the input fields, whereas 'cancel' is not creating a new record in the database	Functional
The links provided on the forms are opening correctly.	Functional
The data feed mechanism for Read and Write files is generating a log with count of entries.	Navigation

Suggested Technical Reading

The project is aimed at making the student understand concepts of Design and Development using IBM Rational tools, WebSphere Application Server and DB2 Database. The following reading reference is easy to understand and should be read to get a clear understanding of capabilities of the tools and how you would leverage them to execute a project.

Technical Reference	URL to access
RAD - Tackling challenges of software development with Rational Application Developer for WebSphere Software	http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html
IBM Education Assistant - Rational Application Developer 7.5	http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html
RSA-Overview of Rational Software Architect for WebSphere Software Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html
Using the new features of UML Modeler in IBM Rational Software Architect Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html
Rational Technical Library	http://www.ibm.com/developerworks/rational/library/