# PROJECT REPORT

**ON**

**Asset Inventory**

**Management**

**BY**

**Rutuja Umrao Jagtap**

**Under the Guidance of**

**Prof. Shivani Budhkar**

**MASTER IN COMPUTER APPLICATION**

**PES MODERN COLLEGE OF ENGINEERING PUNE – 411 005.**

**SAVITRIBAI PHULE PUNE UNIVERSITY 2019 – 20**

**Progressive Education Society’s Modern College of Engineering, Pune-411005**

**CERTIFICATE**

This is to certify that **Rutuja Umrao Jagtap** of Master in Computer Application has successfully completed the Industrial Training Project work titled ‘**Asset Inventory Management**’ during the academic year 2019-20. This report is submitted as partial fulfillment of the requirement of a degree in MCA Engineering of Savitribai Phule Pune University.

|  |  |  |
| --- | --- | --- |
| **Dr. Mrs. K. R. Joshi** | **Dr. V.B. Sangvikar** | **Prof. Shivani Budhkar** |
| **Principal** | **Head of Department** | **Project Guide** |

# ACKNOWLEDGEMENT

I have the great pleasure to present this project on **Asset Inventory Management.** I am thankful to **PES Modern College of Engineering, Shivajinagar, Pune-41005** for giving me the opportunity to carry out the **MCA (Sem VI)** project at their concern.

I am grateful to **Prof. V.B.Sangvikar**, the Head of the Department for her co-operation. My special thanks to **Prof. Shivani Budhkar** for her constructive suggestions and comments and for lending her expertise and talent when needed.

Finally, I would like to thank each and everyone who was directly or indirectly involved with this project report.

**Rutuja Umrao Jagtap**

**Roll No: 53021**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **CONTENTS** | **PAGE NO** |
| **1** | **CHAPTER 1: INTRODUCTION** |  |
|  | 1.1 Company Profile |  |
|  | 1.2 Existing System and Need for System |  |
|  | 1.3 Scope of Work |  |
|  | 1.4 Operating Environment – Hardware and Software |  |
|  | 1.5 Detail Description of Technology Used |  |
| **2** | **CHAPTER 2: PROPOSED SYSTEM** |  |
|  | 2.1 Proposed System |  |
|  | 2.2 Objectives of System |  |
|  | 2.3 User Requirements |  |
| **3** | **CHAPTER 3: ANALYSIS & DESIGN** |  |
|  | 3.1 Entity Relationship Diagram |  |
|  | 3.2 Use Case Diagrams |  |
|  | 3.3 Schema Diagram |  |
|  | 3.4 Activity Diagram |  |

|  |  |  |
| --- | --- | --- |
|  | 3.5 Module Specifications |  |
|  | 3.6 User Interface |  |
|  | 3.7 Table specifications |  |
|  | 3.8 Test Procedures and Implementation |  |
| **4** | **CHAPTER 4: USER MANUAL** |  |
|  | 4.1 User Manual |  |
| **5** | **Conclusion** |  |
| **6** | **Bibliography** |  |

**CHAPTER 1: INTRODUCTION**

Asset inventory is the way an organization lists and provides details of the assets it owns. This can cover a range of different types of assets, from tangible fixed assets such as property and equipment, intangible assets such as intellectual property. Asset inventory management is the way the organization monitors the assets it owns to track and analyze issues such as physical location, maintenance requirements, depreciation, performance, and eventual disposal of the asset.

The Asset Inventory Management project, is a completely customized application especially for the IT Support team in Eternus Solutions Pvt. Ltd. Using this Salesforce application IT Support team gets a broader view of the Assets associations. They are able to keep an easy track on the Software licenses, their subscriptions, and their validity. This application helps to generate various reports related to different kinds of Assets like Desktop, laptops, all related accessories like mouse, keyboard, generic assets like projectors, TV sets, telephones.

This Asset Inventory Management project report contains a deep overview of the project and the working of the application.

# 1.1 COMPANY PROFILE



Company Name:

Eternus Solutions Pvt. Ltd.

Brief About Company:

Eternus Solutions is an IT Consulting Services and outsourcing company providing a range of IT Services to enterprises across various domains globally. Eternus Solutions has carved a niche for itself in the IT industry and cemented its place as India’s leading IT services provider by acquiring elite clientele. The organization has made a mark for itself in the industry in a relatively short span of time through its ability and adherence to commitments to its clients.

Address:

Eternus Solutions Private Limited  
NSG IT Park,  
Sarjaa Rd, Sanewadi, Aundh,  
Pune, Maharashtra 411007  
  
Cell: +91 902-852-8000

# 1.2 EXISTING SYSTEM AND NEED OF SYSTEM

As we all know MS Excel is a mature Microsoft product that has a huge capacity to handle various structured data in various formats.

Here Asset Inventory is also handled using MS Excel, which has its own advantage and disadvantage too. Broadly focusing on the disadvantage of MS Excel,

* Excel makes customer or vendor details hard to maintain over a middle scale organization.
* There arises a frequent problem of data redundancy due to various versions of files.
* As there are multiple collaborators, it becomes tedious to find out manual errors, and also there is a high risk of inconsistent data.
* Even if MS Excel provides various automation support, it does not fulfill the current need of the organization.

Thus to overcome, the organization decided to convert the existing system to a more user-friendly efficient CRM tool.

# 1.3 SCOPE OF THE PROPOSED SYSTEM

Asset Inventory management app helps the IT Support team to manage all the hardware and software assets in the organization. The IT Support team will get a broader view over assets association and the inventory, on various platforms like mobile and desktop.

Asset association

In the organization, every employee is provided with the required equipment by the IT department. The provided assets include desktop, laptops, all the related accessories to desktop and laptop, required software, could be licensed or free.

The IT Support team takes care of the association of the equipment as well as software with the employee.

Using the Asset Inventory management app, the IT team will get a detailed view of the association of the asset in one place.

The user of the application can associate various assets like different on the demand of the employees in the organization.

Assets maintenance

The IT department needs to keep track of the equipment maintenance. So that they are able to replace assets or order for new if required.

The app shows various dynamic reports on the assets’ quality and quantity status. Using those IT departments will know assets availability, it’s working status, their subscriptions.

License tracking

Every employee has the Microsoft license of the organization domain. And every license has its own validity. So on a timely basis, the validity of the license needs to be updated.

Using this app, the IT support team will get timely reminder notifications for the renewal of licenses, and with the help of reports, the status of licenses is tracked easily.

# 1.4 TECHNICAL REQUIREMENT

Minimum requirements are:

* An Octane score of 20,000 or greater
* Network latency of 200 ms or less
* Download speed of 1 Mbps or greater
* At least 5 GB of RAM, with 2 GB available for Salesforce browser tabs

# 1.5 Detail Description of Technology Used

## Salesforce

Salesforce is one of the world's prime cloud computing companies and the number one on-demand customer relationship management(CRM). Salesforce does not need any software installation or hardware or any infrastructure like servers. All we need to access Salesforce is the internet. This empowers even the most non-techie individuals to be able to use the system and configure it as per their needs. Established as Salesforce.com(SFDC) and its customer relationship management (CRM) service and then divided into different sectors like sales cloud, service cloud, community cloud, analytics cloud, data cloud, marketing cloud, app cloud, and so on. Since Salesforce coordinates well with all the platforms and supports all major OS and mobile devices, it is anything but difficult to utilize Salesforce outside of the workplace, thus helps to improve productivity.

Salesforce has a multi-tenant architecture. Multi-tenancy is the fundamental technology utilized as a part of the cloud to share its resources safely and cost-effectively. It's much the same as bank services where various tenants cost-efficiently share a common infrastructure yet safely and with the most protection from other tenants. A cloud utilizes multi-tenant infrastructure to share its assets safely among different applications and occupants (organizations, associations, and so on) that use the cloud. Some clouds utilize virtualization-based architecture to confine occupants; others utilize custom software architecture to take care of business. The multi-tenant outline of a cloud service can dramatically affect the application delivery and the profitability of IT organization

Salesforce.com allows administrators to configure and design systems for complex implementations. Salesforce combines the power of configuration and custom development in its platform i.e. Force.com Platform. This platform user can make use of custom code, workflow rules, approval processes to implement their business logic, and they can integrate the data with other applications, generate reports and do the analytics within no time. The Salesforce CRM model is used in organizations for interactions like emails, meetings, events with customers, and also for prospects like sales, marketing, and support. With Force.com, we can run business in the mobile using the Salesforce1 app. We can build and optimize the apps for mobile using HTML 5 and UI framework and it supports all devices with just one codebase. Salesforce1 downloadable app can be installed from the App Store or Google play on a mobile device.

# CHAPTER 2: PROPOSED SYSTEM

* 1. **PROPOSED SYSTEM**

Asset Inventory Management

With the advent of ever-evolving technology, it becomes necessary to manage IT-related assets efficiently, and managing it the right saves a lot of time. A variety of IT-maintenance software

products exist in the market which can be used to manage IT equipment efficiently. But often these IT-maintenance software products do not always fulfill all the IT requirements of an organization.

For example, most of the IT-maintenance software products available in the market manage

hardware and software assets quite well but fail to capture non-pc data such as telephone system and office material and also lack the ability to support heterogeneous networking in MAC environment. In such cases, developing a good interface to a database where IT information is stored often helps to provide good overviews, good updates, and low maintenance costs.

The project named Asset Inventory Management is a Salesforce-based project especially

built for the IT Support team in Eternus Solution Pvt. Ltd. Asset Inventory Management is a project for the IT Support team where they will be able to maintain the asset inventory under IT department. They will be able to monitor the assets associated with the employees. And also able to manage other assets like telephone, mouse, keyboard, projector etc. IT support will be able to work smoothly with the help of this salesforce based application which is supported on multiple platforms with a 360-degree view around assets.

# OBJECTIVE

The following are the objectives of the proposed system.

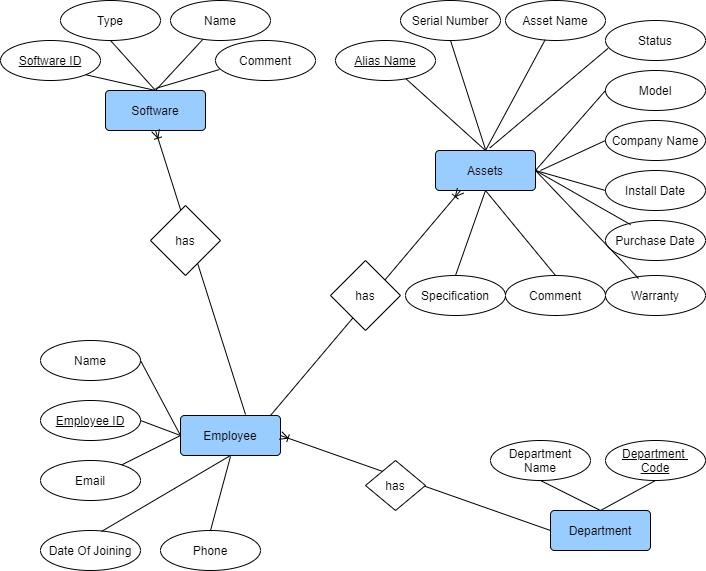
* To manage asset inventory related to the IT Support department.
* To manage various license allocation and its inventory.
* To be able to make proper reports with various filters like date wise, monthly & yearly.

# USER REQUIREMENTS

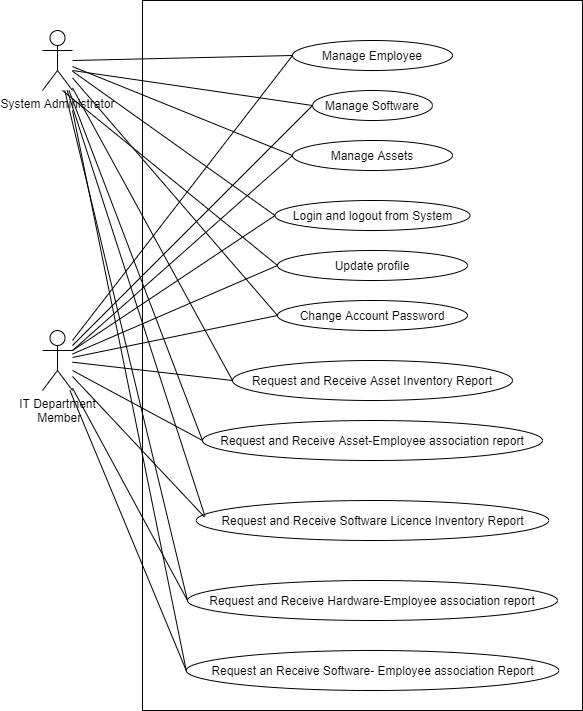
* + The user should have basic knowledge of computers.
  + Users must have proper authorization.
  + Users should have knowledge of the Internet.

# CHAPTER 3: ANALYSIS & DESIGN

**3.1 ER DIAGRAM**

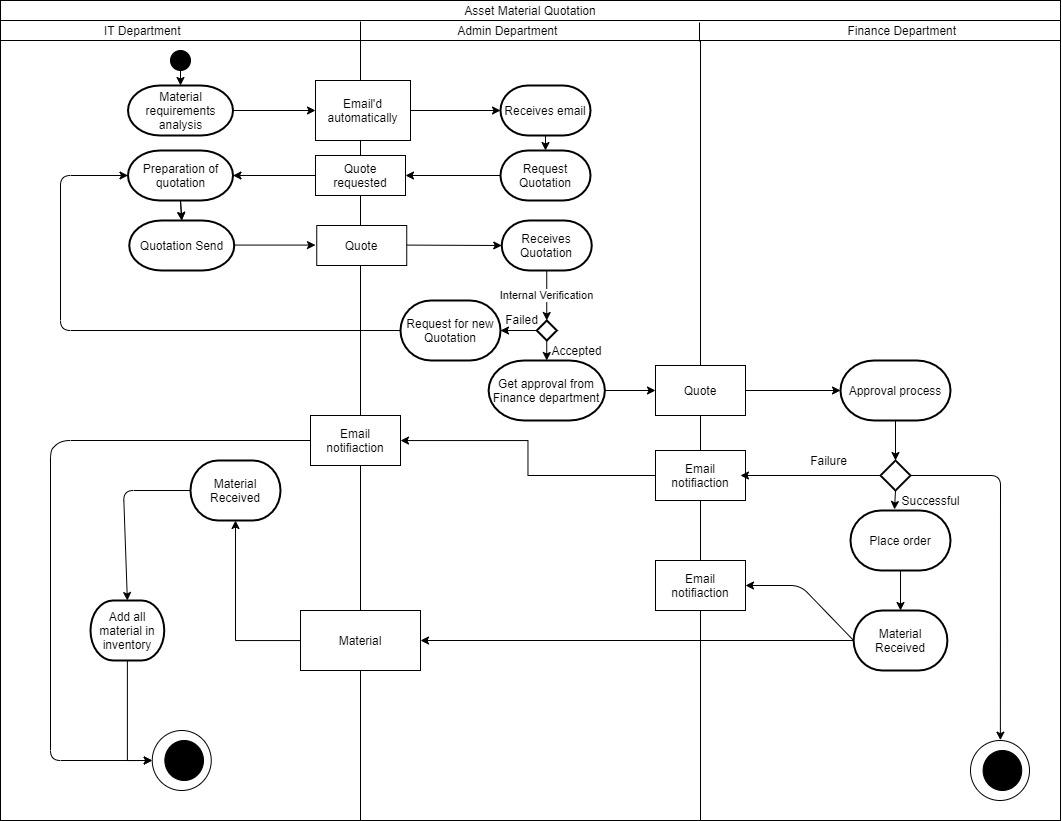


**3.2 USE CASE DIAGRAM**

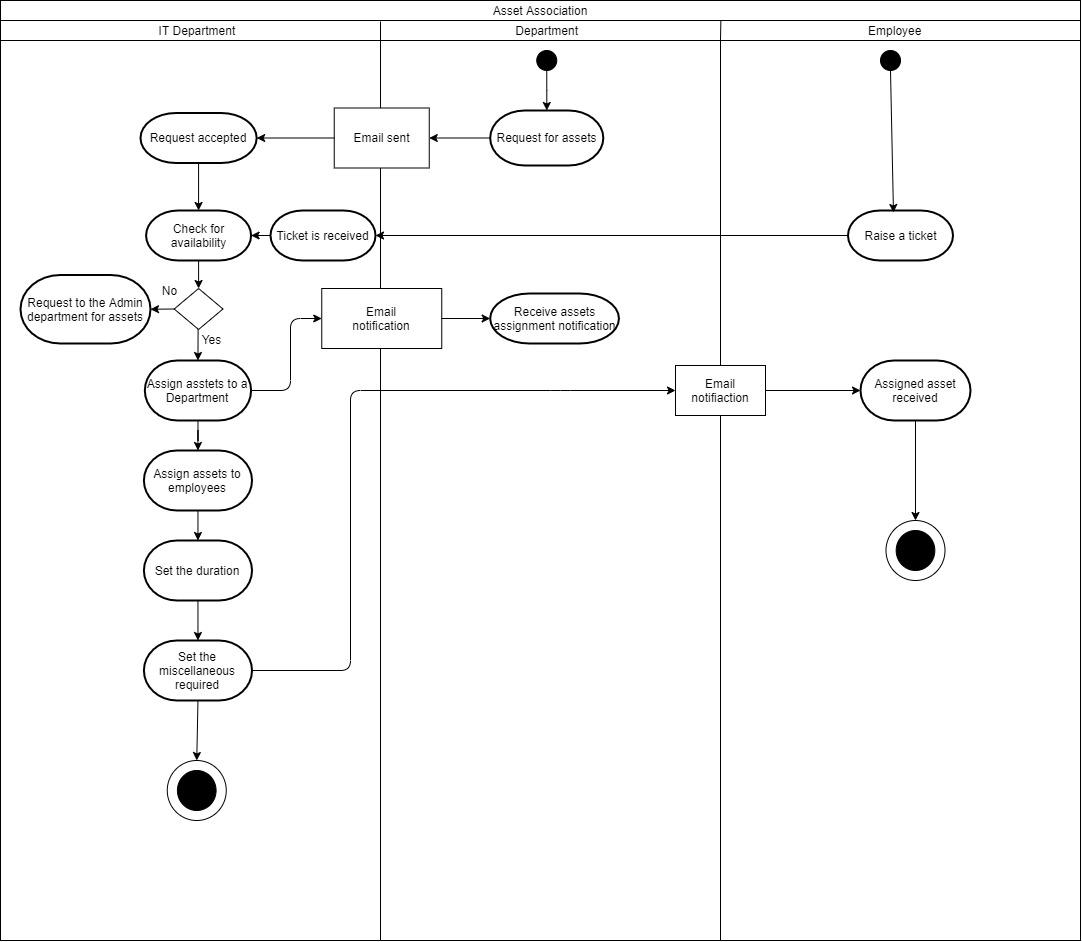
\

**3.3 ACTIVITY DIAGRAM**

**3.3.1 Quotation**



**3.3.2 Asset Assigning**



**3.4 DATA TABLE**

**3.4.1 Asset**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR. NO** | **FIELD LABEL** | **FIELD NAME** | **DATA TYPE** |
| 1 | Asset Level | AssetLevel | Number(9, 0) |
| 2 | Asset Lot | Asset\_Lot\_\_c | Lookup(Asset Lot) |
| 3 | Asset Owner | OwnerId | Lookup(User) |
| 4 | Asset Provided By | AssetProvidedById | Lookup(Department) |
| 5 | Asset Record Type | RecordTypeId | Record Type |
| 6 | Asset Specification | Asset\_Specification\_\_c | Picklist |
| 7 | Asset Type | Asset\_Type\_\_c | Picklist |
| 8 | Company Name | Company\_Name\_\_c | Text(50) |
| 9 | Competitor Asset | IsCompetitorProduct | Checkbox |
| 10 | Created By | CreatedById | Lookup(User) |
| 11 | Department | AccountId | Lookup(Department) |
| 12 | Description | Description | Long Text Area(32000) |
| 13 | Employee | ContactId | Lookup(Employee) |
| 14 | ESPL Asset Number | Name | Text(255) |
| 15 | External Id | ExternalIdentifier | Text(255) |
| 16 | Install Date | InstallDate | Date |
| 17 | Last Modified By | LastModifiedById | Lookup(User) |
| 18 | Manufacture Date | ManufactureDate | Date |
| 19 | Model Number | Model\_Number\_\_c | Text(20) |
| 20 | Price | Price | Currency(18, 0) |
| 21 | Serial Number | SerialNumber | Text(80) |
| 22 | Specific Asset Type | Specific\_Asset\_Type\_\_c | Picklist |
| 23 | Status | Status | Picklist |
| 24 | Status Reason | StatusReason | Picklist |
| 25 | Usage End Date | UsageEndDate | Date |
| 26 | Vendor Name | Vendor\_Name\_\_c | Formula (Text) |

**.4.2 Asset Lot**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR. NO** | **FIELD LABEL** | **FIELD NAME** | **DATA TYPE** |
| 1 | Allotted Asset Count | Allotted\_Asset\_Count\_\_c | Number(18, 0) |
| 2 | Asset Lot Number | Name | Text(80) |
| 3 | Asset Type | Asset\_Type\_\_c | Picklist |
| 4 | Created By | CreatedById | Lookup(User) |
| 5 | Defective Asset Count | Defective\_Asset\_Count\_\_c | Number(18, 0) |
| 6 | Invoice Number | Invoice\_Number\_\_c | Text(30) |
| 7 | Last Modified By | LastModifiedById | Lookup(User) |
| 8 | Model Number | Model\_Number\_\_c | Text(20) |
| 9 | Not Allotted Asset Count | Not\_Allotted\_Asset\_Count\_\_c | Number(18, 0) |
| 10 | Purchase Date | Purchase\_Date\_\_c | Date |
| 11 | Quantity | Quantity\_\_c | Number(18, 0) |
| 12 | Specific Asset Type | Specific\_Asset\_Type\_\_c | Picklist |
| 13 | Usage End Date | Usage\_End\_Date\_\_c | Date |
| 14 | Vendor Name | Vendor\_Name\_\_c | Master-Detail(Vendor) |

**3.4.3 Department**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR. NO** | **FIELD LABEL** | **FIELD NAME** | **DATA TYPE** |
| 1 | Active | Active\_\_c | Picklist |
| 2 | Billing Address | BillingAddress | Address |
| 3 | Clean Status | CleanStatus | Picklist |
| 4 | Created By | CreatedById | Lookup(User) |
| 5 | Department Code | AccountNumber | Text(40) |
| 6 | Department Name | Name | Name |
| 7 | Department Owner | OwnerId | Lookup(User) |
| 8 | Description | Description | Long Text Area(32000) |
| 9 | Employees | NumberOfEmployees | Number(8, 0) |
| 10 | Fax | Fax | Fax |
| 11 | Industry | Industry | Picklist |
| 12 | Last Modified By | LastModifiedById | Lookup(User) |
| 13 | Phone | Phone | Phone |
| 14 | Website | Website | URL(255) |

**3.4.4 Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR. NO** | **FIELD LABEL** | **FIELD NAME** | **DATA TYPE** |
| 1 | Birthdate | Birthdate | Date |
| 2 | Created By | CreatedById | Lookup(User) |
| 3 | Date Of Joining | Date\_Of\_Joining\_\_c | Date |
| 4 | Date of Resignation Or De-Activation | Date\_of\_Resignation\_Or\_De\_Activation\_\_c | Date |
| 5 | Deactive | Deactive\_\_c | Checkbox |
| 6 | Department | Department | Text(80) |
| 7 | Department Name | AccountId | Lookup(Department) |
| 8 | Description | Description | Long Text Area(32000) |
| 9 | Email | Email | Email |
| 10 | Employee ID | Employee\_ID\_\_c | Text(30) |
| 11 | Employee Owner | OwnerId | Lookup(User) |
| 12 | Fax | Fax | Fax |
| 13 | Home Phone | HomePhone | Phone |
| 14 | Last Modified By | LastModifiedById | Lookup(User) |
| 15 | Last Stay-in-Touch Request Date | LastCURequestDate | Date/Time |
| 16 | Mailing Address | MailingAddress | Address |
| 17 | Microsoft Licence | Microsoft\_Licence\_\_c | Lookup(Asset) |
| 18 | Microsoft Licence Type | Microsoft\_Licence\_Type\_\_c | Picklist |
| 19 | Mobile | MobilePhone | Phone |
| 20 | Name | Name | Name |
| 21 | Phone | Phone | Phone |
| 22 | Reports To | ReportsToId | Lookup(Employee) |
| 23 | Title | Title | Text(128) |

**Vendor**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR. NO** | **FIELD LABEL** | **FIELD NAME** | **DATA TYPE** |
| 1 | Address | Address\_\_c | Text Area(255) |
| 2 | Asset Lot Count | Asset\_Lot\_Count\_\_c | Roll-Up Summary (COUNT Asset Lot) |
| 3 | Created By | CreatedById | Lookup(User) |
| 4 | Defective Asset Count | Defective\_Asset\_Count\_\_c | Roll-Up Summary (SUM Asset Lot) |
| 5 | Email | Email\_\_c | Email |
| 6 | Last Modified By | LastModifiedById | Lookup(User) |
| 7 | Owner | OwnerId | Lookup(User,Group) |
| 8 | Phone | Phone\_\_c | Phone |
| 9 | Vendor Name | Name | Text(80) |
| 10 | Website | Website\_\_c | URL(255) |

* 1. **TEST PROCEDURES AND IMPLEMENTATION**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation. It is a process of executing a program with a primary objective of finding errors. Testing gives the guarantee that the software does not fail and runs according to its specifications and in the way the end-user expects. This can be done by various software testing techniques that provide systematic guidance for designing tests that exercise the internal logic of software components and exercise the input and output domains of the program to uncover errors in program function, behavior, and performance.

## Testing Objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

* Testing is a process of executing a program with the intent of finding an error.
* A successful test is one that uncovers an as yet undiscovered error.
* The tests are inadequate to detect possibly present errors.
* The software more or less conforms to the quality and reliability standards.

The following software testing techniques were used in order to uncover errors:

1. White-box testing
2. Black box testing
3. Validation testing
4. GUI testing
5. Performance testing
6. Regression testing
7. Unit testing
8. Integration testing

## Above mentioned techniques are described in short below:

1. **White-box testing**

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of software testing that tests internal structures or workings of an application. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases.

## Black box testing

Black box testing, also called behavioral testing, focuses on the functional requirements of the software. It is related to input and output only and not related to the internal structure of the program. This testing was also done so as to find errors such as:

* + Initialization and termination errors
  + Behavior and performance errors
  + Incorrect or missing functions
  + Errors in data structures and external database access

## Validation Testing

Validation testing occurs when we have to declare some validation regarding our input screens

## GUI Testing

GUI test verifies the alignment, font, images, and buttons, as per the requirements. GUI Testing is also called Look and Feel Testing.

## Performance Testing

Performance requirements are expressed in terms of time required for response time for accessing the features of the software.

## Unit Testing

Unit testing is a test that tests every single module of the software to check

for errors. This is mainly done to discover errors in the code of the services that are uploading data on the cloud.

The main goal of the unit testing would be to isolate each part of the program and to check the correctness of the code.

## Integration Testing

The main purpose of the integration testing is to test the functional and performance requirements of the major items of the project. All the modules of the project developed individually would be combined together and tested as a whole system in the integration testing. In Integration Testing, the individual software modules are combined and tested as a whole unit. The integration testing generally follows unit testing where each module is tested as a separate unit.