

P-3569



# TRACK ME SOFTWARE AS SERVICE

## PROJECT REPORT

*Submitted by*

**P.NATARAJAN**

**Register No: 0920108013**

*In partial fulfillment for the award of the degree of  
of*

**MASTER OF ENGINEERING**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**KUMARAGURU COLLEGE OF TECHNOLOGY**

**(An Autonomous Institution Affiliated to Anna University of Technology, Coimbatore)**

**COIMBATORE – 641 049**

**APRIL 2011**

3569

**KUMARAGURU COLLEGE OF TECHNOLOGY****(An Autonomous Institution Affiliated to Anna University of Technology, Coimbatore)****COIMBATORE – 641 049****Department of Computer Science and Engineering****PROJECT WORK****PHASE II****APRIL 2011**

This is to certify that the project entitled

**TRACK ME SOFTWARE AS SERVICE**

is the bonafide record of the project work done by

**P. NATARAJAN****Register No: 0920108013**

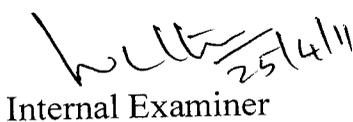
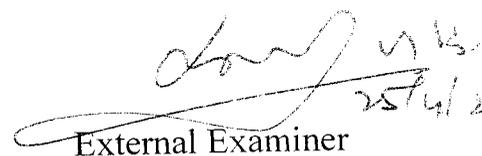
of M.E. (Computer Science and Engineering) during the year 2010-2011.



Project Guide

  
Head of the Department

Submitted for the Project Viva-Voce examination held on

25/4/11  
Internal Examiner  
External Examiner

## DECLARATION

I affirm that the project work titled “**TRACK ME SOFTWARE AS SERVICE**” being submitted in partial fulfillment for the award of M.E degree is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

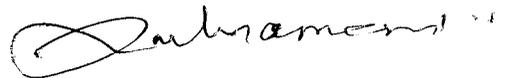


P.Natarajan

0920108013

I certify that the declaration made above by the candidate is true

Signature of the Guide,



**Mr. V. Subramani M.Tech.,**  
**Associate Professor**

Department of Computer Science & Engineering,  
Kumaraguru College of Technology,  
(An Autonomous Institution)  
Coimbatore.

## ACKNOWLEDGEMENT

First and foremost, I would like to thank the almighty for enabling me to complete this project.

I whole-heartedly wish to express my special thanks to our Chairman, Management, and Director for giving this great opportunity to pursue this course. I thank **Dr. S. Ramachandran**, *Principal*, for providing me with the necessary facilities and Infrastructure to work on this project.

I would like to extend my thanks to **Mrs. P. Devaki, M.E., (Ph.D)**, *HOD*, Department of Computer Science and Engineering for her valuable suggestions.

I express my heartiest thanks to my Project Guide **Mr.V.Subramani M.Tech.**, *Associate Professor*, Department of Computer Science and Engineering for his valuable suggestions and support right from the beginning to perform my project work extremely well.

I thank our Project Coordinators **Mr.V.Subramani M.Tech.**, *Associate Professor* and **Mrs.V.Vanitha M.E., (Ph.D)**, *Associate Professor*, Department of Computer Science and Engineering, for their valuable advice and guidance throughout this project.

I would like to convey my honest thanks to all **Teaching and Non-Teaching** staff members of the department for their support.

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>Abstract</b>	<b>vii</b>
	<b>Abstract (Tamil)</b>	<b>viii</b>
	<b>List of Figures</b>	<b>ix</b>
	<b>List of Abbreviations</b>	<b>ix</b>
<b>1</b>	<b>Introduction</b>	<b>1</b>
	1.1 Cloud Computing	1
	1.1.1 Types of Clouds	1
	1.1.2 Cloud Computing Architecture	2
	1.2 Service Oriented Architecture	3
	1.3 Virtualization	5
<b>2</b>	<b>Literature Survey</b>	<b>9</b>
<b>3</b>	<b>System Analysis</b>	
	3.1 Drawbacks of Existing System	11
	3.2 Proposed System	12
<b>4</b>	<b>System Specification</b>	
	4.1 Hardware Requirements	14
	4.2 Software Requirements	14
	4.3 Software Description	14
<b>5</b>	<b>Project Description</b>	
	5.1 Problem Definition	19
	5.2 Overview of the Project	19
	5.3 Module Description	19

<b>6</b>	<b>Results and Discussion</b>	22
<b>7</b>	<b>Conclusion &amp; Future Outlook</b>	23
<b>8</b>	<b>Appendices</b>	24
	8.1 Source Code	24
	8.2 Screen Shots	41
<b>9</b>	<b>References</b>	46

NBA Accredited, ISO Certified Autonomous Institution  
Affiliated to Anna University of Technology, Coimbatore  
KSR Kalvi Nagar, Tiruchengode-637 215

# DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL CONFERENCE

ON

RECENT TRENDS AND ADVANCES IN INFORMATION TECHNOLOGY - NCRAIT '11

This is to certify that Mr./Ms. **P. NATARAJAN**.....

of **Kumaraguru...college...of...Technology**..... has participated

and presented a paper on **A. New...Approach...to...EHR...wing**.....

on **Cloud...computing**.....in the National Conference on Recent Trends

and Advances in Information Technology (NCRAIT '11) held on

05th March 2011.

NCRAIT '11

Dr. S. Govindarajan



CONVENOR

H O D

PRINCIPAL

## **ABSTRACT**

Cloud computing one of the emerging fields in information technology is gaining popularity as means of saving cost of IT ownership. Cloud Computing is Internet based computing, whereby shared resources, software and information, are provided to computers and devices on-demand, like the electricity grid and it is also based on several other computing areas like virtualization, Service Oriented Architecture and grid computing. Due to its accelerating time to market ready-to-use, dynamically scalable computing infrastructure and software services it has gained popularity among the IT organizations. Cloud computing is recognized as a potentially viable way of encouraging adoption of the next generation of standardized interoperable systems. A new cloud based approach is proposed in this paper, which helps to overcome the short comings in Customer Relationship Management. This approach promotes a common way for the interoperability of systems in device independent manner. It will be very useful for the IT organizations and other organizations for maintaining a good relationship with the customers using the device independent software providing it as a service.

## ஆய்வுச்சுருக்கம்

கணிணி மேகம் மூலம் நாம் பல்வேறு வேலைகளை எளிதாக செய்யலாம். குலுவாக செயல்படுபவர்களுக்கு 'கணிணி மேகம்' அதிக பயனுள்ளது. எந்த ஒரு பணியையும் ஒருவர் மற்றவரை நேரில் சந்தித்து எந்த கூட்டம் போட்டு எடுத்துக் கூற வேண்டிய அவசியமில்லை. அவர்களுக்கு அணத்து தகவல்களையும் இணையம் வழியாகத் தரலாம். இருக்கும் ஆவணங்களைப் உடனுகுடன் அவ்வப்போது இருக்கும் மாற்றங்களை உடனுகுடன் காணும்படிச் செய்யலாம். ஒரு நிறுவனதிற்குள் மட்டுமில்லாமல், ஒரு நாட்டிற்குள் மட்டுமில்லாமல், இந்த குழு உலகம் முழுவதும் பரந்து விரிந்திருந்தாலும் இது சாத்தியமாகும். இதில் சேவையாக மென்பொருள் நிறுவப்படும்.

## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
1.1	Cloud Computing Architecture	2
1.2	Layers of Cloud Computing	3
1.3	Virtualization	8
1.4	Project Architecture	12
1.5	Visualforce system architecture	18

## LIST OF ABBREVIATIONS

SAAS	Software as a service
PAAS	Platform as a service
IAAS	Infrastructure as a service
SOA	Service Oriented Architecture

## CHAPTER 1

### INTRODUCTION

#### 1.1 CLOUD COMPUTING

Cloud computing is service of IT (Information Technology) infrastructure and managed sharing of IT resources. Instead of running your applications yourself, they run on a shared data center. When you use any application that runs in the cloud, you just login, customize it, and start using it. The benefits of moving to the Cloud includes reduced cost of ownership, easy integration to the other systems, unrestricted access to the application with just an internet connection and reduced or almost zero maintenance cost. Service Oriented Architecture (SOA) and Virtualization together forms the cloud computing.

##### 1.1.1 TYPES OF CLOUDS

Basically there are three types of clouds. They are:

- a) Private Clouds
- b) Public Clouds
- c) Hybrid Clouds

###### 1.1.1.1 Private Clouds

**Private cloud** is set up and managed by an enterprise's own IT department and run inside the organization firewall. If any organization has large number of user and resources, then organization hosts cloud computing within their own firewall. This type of cloud computing is dedicated to that organization. It does not share any resource outside their organization. Any big organization like AT&T, Verizon or Bank of America open their infrastructure or data center near to low cost area and makes the service available through their own cloud, then it called as private Cloud computing. It shares some benefit like efficiency, High availability, elastic capacity, lower cost over time, full access and flexibility, direct control over quality, service and security.

###### 1.1.1.2 Public Clouds

**Public cloud** is offering service by third party vendor over internet. If any vendor

in public cloud type. This type of cloud shares some benefit like efficiency, High availability, elastic capacity, Low upfront cost, less or no hardware setup and less or no system management. This type of cloud computing service is provided by Amazon EC2, Microsoft Azure, Sun Microsystems cloud, Salesforce etc.

### 1.1.1.3 Hybrid Cloud

**Hybrid cloud** is combination of any or all of the other types of cloud. This type of cloud is gaining lot of popularity among organizations. This type of cloud computing gives organization more flexibility to manage and share resource between private and public cloud. Like if any organization host their application in public cloud and during peak sales time they need more server and space to handle this request, they can go for public cloud. In this type of cloud computing Model Company keeps all sensitive data (transaction or credit card data) in private cloud and less sensitive data in public cloud. It shares benefits like efficiency, High availability, elastic capacity, more control over quality, service and security, less cost over time compare to public and community cloud.

### 1.1.2 Cloud Computing Architecture:

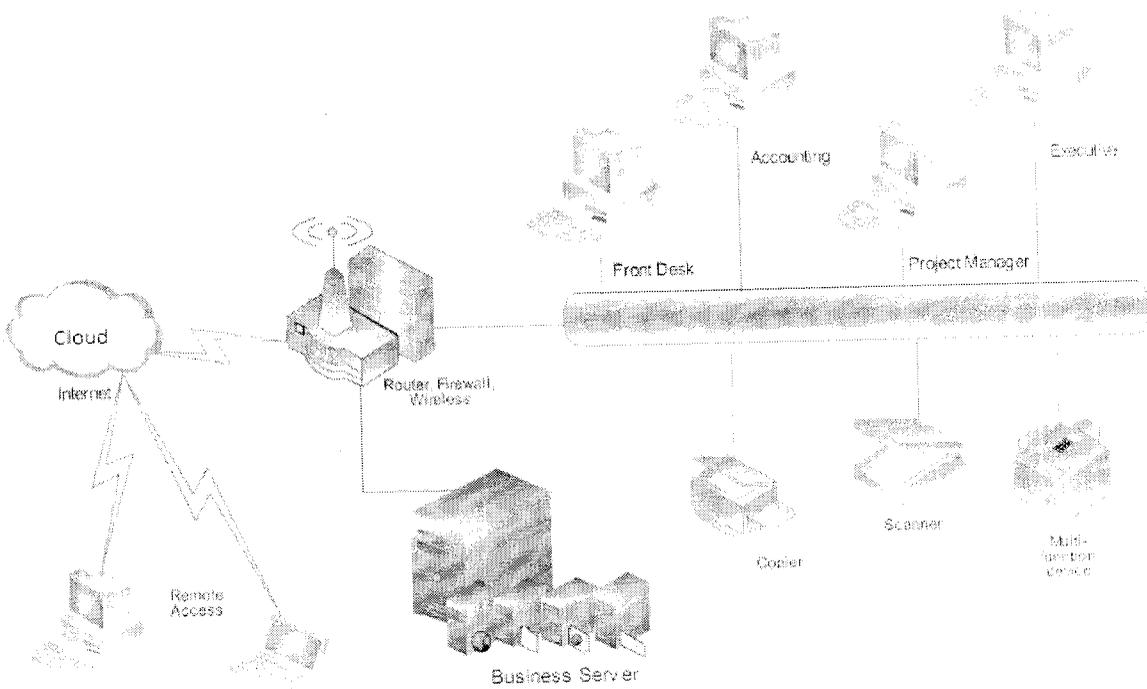


Figure. 1.1 Cloud Computing Architecture

### 1.1.3 Layers of cloud computing:

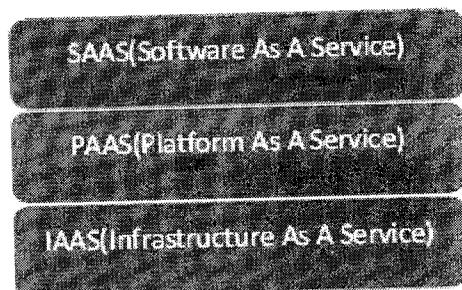


Figure 1.2 Layers of cloud computing

**SAAS (Software As A Service)** - This is the top technology layer of Cloud Computing and oldest among these three. Under this layer organization gets fully functional applications on-demand to provide specific services such as email management, CRM, ERP, web conferencing and an increasingly wide range of other applications. These software licenses are managed by Cloud computing company.

**PAAS (Platform As A Service)** – Second layer of cloud computing is PAAS (Platform As A Service). In this layer organization gets mostly an operating environment to develop application, to run application or to deploy application. PAAS provides operating environment like Java, J2EE, .Net, Window, Linux etc.

**IAAS (Infrastructure AS A Service)** – This layer provides all basic, physical and virtual resources used in application for any organization. This includes virtual platform (space on server) on which required operating environment and application are deployed. It also includes storage and datacenter.

## 1.2 Service Oriented Architecture

Service Oriented Architecture is the evolution of a system or software architecture which ensures reusability, extensibility and flexibility. It is also a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. Service is a mechanism to enable access to one or more capabilities where the

and policies as specified by the service description. In SOA approach to modeling software, software is organized in terms of services each capturing reusable functionality that can be discovered at run time and accessed in location transparent manner through well-defined open standards technology based interfaces.

**Modular and loosely coupled** - SOA cuts big monolithic systems and services into small modular services. SOA not only disengages the process from system but it also makes loose couple among systems. If, for instance, a SOA implemented organization wants to implement a new service (Internal service, External service or Cloud based services), it can just plug that service or remove that service from the organization system without touching any existing system.

**Business Driven** - Now a days, business changes very fast. Mergers and Acquisitions are very common phenomena. Business needs are also changes frequently. There are different permutations and combination coming in market. IT also has to synchronize with all these business permutations and combination. SOA gives flexibility to work with all these changes with minimal effort. SOA is very close to business as well as the business people. One can define or change business process to address business needs in very small time with minimal effort. In short, organization business can change quickly along with the fast changing market.

**Platform Independent** - SOA is completely platform independent. SOA service client or consumer can use SOA service without knowing SOA service provider platform, language or operating system. So, one can write SOA service without worrying about who is going to consume the service.

**Easy Service enablement** - There was big challenge for any organization to make any change in existing legacy system. SOA provides an easy tool to enable existing monolithic application or legacy mainframe system into a SOA service without knowing anything about those systems. Through SOA tool, one can expose interface(s) from existing system as service and use this service in other applications without touching any existing functionality.

**Low Cost development and maintenance** - As I stated earlier, SOA is modular and loosely coupled. So one can implement new service or modify existing service(s) without touching

change in application that uses SOA. SOA also provides the flexibility, such that service can be reused in any other application. For example, if an organization is using third party service (SAAS) for an application, it can reuse this service in other applications without investing on license, server space or maintenance. In other words, if organization builds a service for one application, it can reuse this service in multiple applications.

**Easy learning curve** - SOA is modular and loosely coupled, as such, it has a very easy learning curve. One can start to work on any service without knowing the whole application. SOA technology is completely based on XML and it is one of the easiest technologies and it is accepted by all applications and software.

**Increased operational efficiency** - To reuse existing SOA service, one can create new service by using current system or service. In summary, SOA helps us in creating and delivering a new product quickly.

### **1.2.1 Common principles of service-orientation:**

- services are loosely coupled
- services share a formal contract
- services abstract underlying logic
- services are composable
- services are reusable
- services are autonomous
- services are stateless
- services are discoverable

#### **Services are Loosely Coupled**

Coupling refers to a connection between two things..A fundamental principle of service-orientation is that units of logic that are classified as services retain a minimal level of coupling

#### **Services Share a Formal Contract**

Units of automation logic that are classified as services must provide a contract in which the

description documents that comprise the technical service contract and provide published details about the service, such as its programmatic interface, communication requirements, constraints, properties, usage policies, and even preferences.

### **Service are Composable**

The principle of composability applies to individual services, and strongly encourages that services be designed in support of aggregated assembly as composition controllers, members, or both. By ensuring that services are capable of participating in multiple compositions, an inventory of adaptive services can be accumulated.

### **Services are Autonomous**

Autonomy represents the governance a service has at the time of execution over the underlying application logic required to carry out the functions exposed by the service contract. The extent to which a service can control its underlying logic dictates the level of its autonomy.

### **Services are Stateless**

Maintaining a condition of statelessness therefore benefits a service by increasing its scalability and availability. Furthermore, the processing of state information typically requires automation logic that is specific to the business task being executed. For services to maximize reuse potential, their context and underlying logic must be as generic (task-neutral) as possible.

#### **1.2.2 Service-oriented architectures have the following key characteristics:**

- SOA services have self-describing interfaces in platform-independent XML documents. Web Services Description Language (WSDL) is the standard used to describe the services.
- SOA services communicate with messages formally defined via XML Schema (also called XSD). Communication among consumers and providers or services typically happens in heterogeneous environments, with little or no knowledge about the provider. Messages between services can be viewed as key business documents processed in an

- SOA services are maintained in the enterprise by a registry that acts as a directory listing. Applications can look up the services in the registry and invoke the service. Universal Description, Definition, and Integration (UDDI) is the standard used for service registry.
- Each SOA service has a quality of service (QoS) associated with it. Some of the key QoS elements are security requirements, such as authentication and authorization, reliable messaging, and policies regarding who can invoke services.

### 1.3 Virtualization

Virtualization is the partitioning of not only mainframe server but any physical server into multiple virtual servers. It gives organization maximum utilization of hardware with same CAPEX (ongoing capital expenditure) and OPEX (ongoing Operational expenditure). Each server acts like a real physical server that can run on operating system with just like physical server. Now companies are partitioning their physical server into multiple virtual servers and run their application on virtual servers with same resources and less expenditure

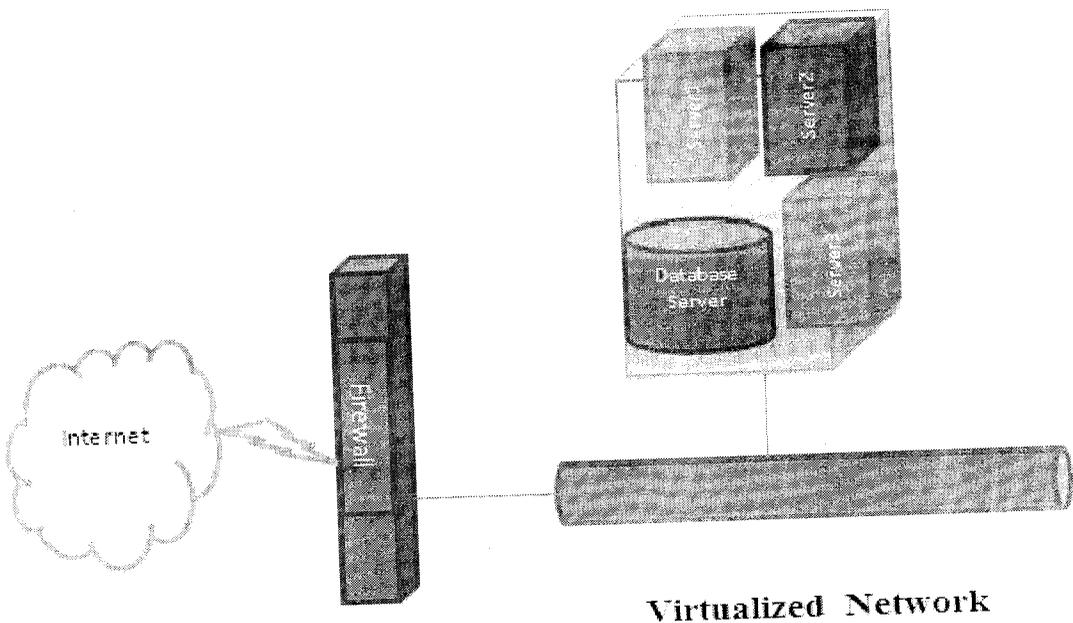


Figure. 1.3. Virtualization

There are three different types of virtualization

**Desktop virtualization** – Desktop virtualization allow us to run different desktop for different users simultaneously on the same hardware.

**Storage virtualization** – Storage virtualization is the pooling of physical storage from multiple network devices on the same hardware.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 CLOUD COMPUTING:

Cloud computing in the world of Information Technology refers to the Internet-based Computing. In other words Cloud Computing is “Everything as a Service, Anytime, Anywhere”. Instead of running your applications yourself, they run on a shared data center. When you use any application that runs in the cloud, you just login, customize it, and start using it. The benefits of moving to the Cloud includes reduced cost of ownership, easy integration to the other systems, unrestricted access to the application with just an internet connection and reduced or almost zero maintenance cost.

**SAAS:** Delivery of software as a service. A single copy of software can be made available to consumers on demand as a shared service accessible over remote network location and charged on subscription or pay-per-use basis. The term SaaS denotes application software provided in such a mode.

#### EXSISTING SYSTEM AND RELATED WORKS:

No previous systems were existing in this model since this is an emerging trend in Information technology currently. Previously the mainframe systems are in existence in which the user has to wait for until the system completes one job. They are device dependent and its more costly to use the system.

##### 2.1.1 Shane Robison [1]:

Cloud computing in the world of Information Technology refers to the Internet-based Computing. In other words Cloud Computing is “Everything as a Service, Anytime, Anywhere”. Instead of running your applications yourself, they run on a shared data center.

Advantages:

- Reduced cost
- Instant updates

##### 2.1.2 Tek Tsai, Xin Sun, Janaka Balasooriya[2]:

Cloud computing is a set of enabling technologies which services bigger and flexible

Computing and SOA will co-exist, complement and support each other. SOA is also a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.

### **2.1.3 Bhaskar Prasad Rimal, Eunmi Choi, Ian Lumb[3]:**

Cloud computing architecture is the design of software solutions that uses internet-accessible on demand-service. The services are accessible anywhere in the world. Basically there are three different types of cloud. Public cloud, Private cloud and hybrid cloud.

### **2.1.4 V.Choudary[4]:**

A single copy of software can be made available to consumers on demand as a shared service accessible over remote network location and charged on subscription or pay-per-use basis. The term SaaS denotes application software provided in such a mode. SaaS commonly referred to as Application Service Provider (ASP).

### **2.1.5 Jianfeng Yang, Zhibin Chen[5]:**

There are more security issues which should be addressed while moving to the cloud computing. In this paper they have clearly said about the issues in Cloud computing. Security is only the disadvantage which makes all stand behind the Cloud Computing technology.

### **2.1.7 Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen and Zhenghu Gong[7]:**

Cloud computing is based on several other computing research areas such as HPC, virtualization, utility computing and grid computing. The cloud computing has its own conceptual, technical, economic and user experience characteristics. The service oriented, loose coupling, strong fault tolerant, business model and ease use are main characteristics of cloud computing. Clear insights into cloud computing will help the development and adoption of this evolving technology both for academe and industry.

### **2.1.8 Junjie Peng, Xuejun Zhang, Zhou Lei, Bofeng Zhang[8]:**

A detailed introduction and comparison of several popular cloud platforms is presented in this paper. From the analysis and comparison, users will be more clear to make their decisions

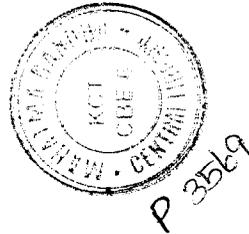
## CHAPTER 3

### SYSTEM ANALYSIS

#### 3.1 DRAWBACKS OF EXISTING SYSTEM:

##### 3.1.1 SOFTWARE

- No better performance
- No Instant updates
- Limited storage capacity
- Decreased data reliability
- No Universal document access
- Latest version unavailable
- Device dependent



#### 3.2 PROPOSED SYSTEM:

In fast running business world its difficult to install and maintain softwares in all systems individually. Every organisation containing many branches which are locate at distinct places. For effective use of the information we need all the information together which are located at different places at right time in right format.

In proposed approach , System takes place on

- ✓ Software On-demand basis
- ✓ Also getting real time data

#### TrackMe Sytem:

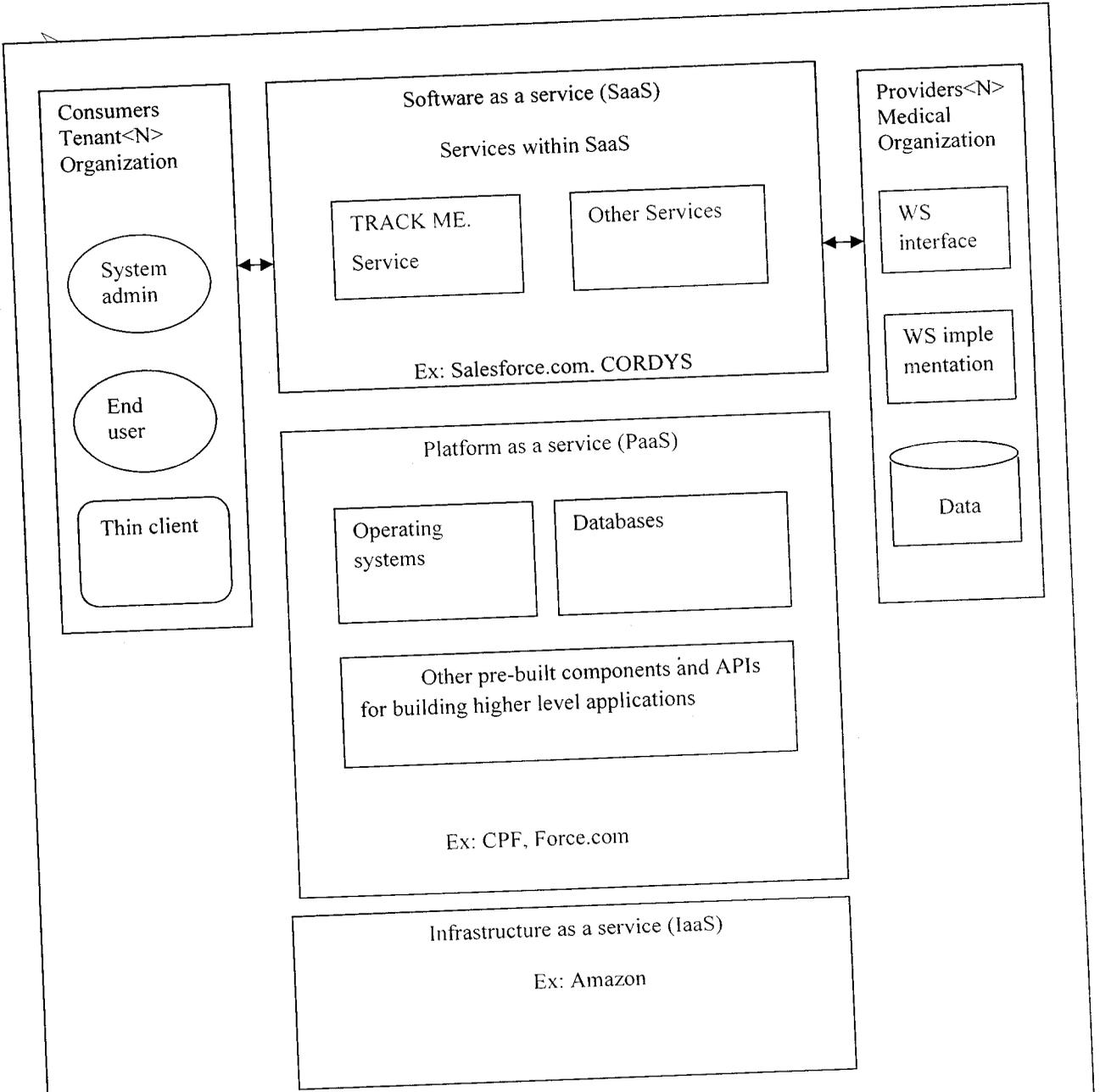
#### Problem Overview:

- In today's extremely challenging business environment, every organisation **measuring their success** by easy access to data anywhere and anytime.

- We need develop the software as service so that it can be accessed anywhere in device independent manner.

**Challenge:**

- Developing the software as service is **highly valuable** but also it is **technically challenging**.



**End User:**

- Access the system as a service
- Reporting to know the business

**Expected Advantage of this Proposed System:**

- ✓ Better performance
- ✓ Instant updates
- ✓ Unlimited storage capacity
- ✓ Increased data reliability
- ✓ Universal document access
- ✓ Latest version available
- ✓ Device independence
- ✓ Real time data is retrieved
- ✓ On-demand data

## CHAPTER 4

### SYSTEM SPECIFICATION

#### 4.1 HARDWARE REQUIREMENTS

Processor	: Pentium III and above
Clock speed	: 550MHz
Hard Disk	: 20GB
RAM	: 128MB or above
Cache Memory	: 512KB
Operating System	: All Operating Systems

#### 4.2 SOFTWARE REQUIREMENTS

Eclipse with Force.com IDE

Mozilla Firefox or Internet Explorer 6 or 7 (Most Compatible)

#### 4.3 SOFTWARE DESCRIPTION:

##### **Force.com IDE:**

- ✓ Development environment with comprehensive application development features, including code editors, and programming languages for optimum productivity.

##### **APEX:**

Force.com Apex code is a strongly-typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Force.com platform server in conjunction with calls to the Force.com API. Using syntax that looks like Java and acts like database stored procedures, Apex code enables developers to add business logic to most system events, including button clicks, related record updates, and Visualforce pages. Apex scripts can be initiated by Web service requests and from triggers on objects.

As a language, Apex is:

### **Integrated**

Apex provides built-in support for common Force.com platform idioms, including:

- Data manipulation language (DML) calls, such as INSERT, UPDATE, and DELETE, that include built-in DmlException handling
- Inline Salesforce.com Object Query Language (SOQL) and Salesforce.com Object Search Language (SOSL) queries that return lists of sObject records
- Looping that allows for bulk processing of multiple records at a time
- Locking syntax that prevents record update conflicts
- Custom public Force.com API calls that can be built from stored Apex methods
- Warnings and errors issued when a user tries to edit or delete a custom object or field that is referenced by Apex

Apex is based on familiar Java idioms, such as variable and expression syntax, block and conditional statement syntax, loop syntax, object and array notation, pass by reference, and so on. Where Apex introduces new elements, it uses syntax and semantics that are easy to understand and encourage efficient use of the Force.com platform. Consequently, Apex produces code that is both succinct and easy to write.

### **Data focused**

Apex is designed to thread together multiple query and DML statements into a single unit of work on the Force.com platform server, much as developers use database stored procedures to thread together multiple transaction statements on a database server. Note that like other database stored procedures, Apex does not attempt to provide general support for rendering elements in the user interface.

### **Rigorous**

Apex is a strongly-typed language that uses direct references to schema objects such as object and field names. It fails quickly at compile time if any references are invalid, and stores all custom field, object, and class dependencies in metadata to ensure they are not deleted while required by active Apex scripts.

### **Hosted**

Apex is interpreted, executed, and controlled entirely by the Force.com platform.

### **Multitenant aware**

Like the rest of the Force.com platform, Apex runs in a multitenant environment.

Apex is designed to guard closely against runaway scripts.

preventing them from monopolizing shared resources. Any scripts that violate these limits fail with easy-to-understand error messages.

### Automatically upgradeable

Apex never needs to be rewritten when other parts of the Force.com platform are upgraded. Because the compiled code is stored as metadata in the platform, it always gets automatically upgraded with the rest of the system.

### Easy to test

Apex provides built-in support for unit test creation and execution, including test results that indicate how much code is covered, and which parts of your code could be more efficient. Salesforce.com ensures that scripts always work as expected by executing all unit tests stored in metadata prior to any platform upgrades.

All Apex runs entirely on-demand on the Force.com platform, as shown in the following architecture diagram:

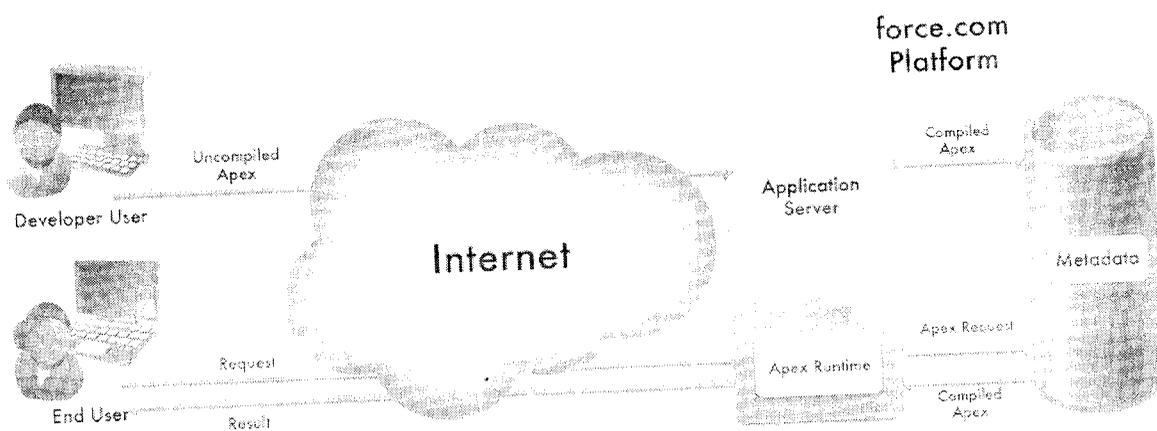


Figure 1.5: Apex code is compiled, stored, and run entirely on the Force.com platform. When a developer writes and saves an Apex script to the platform, the platform application server first compiles the code into an abstract set of instructions that can be understood by the Apex runtime interpreter, and then saves those instructions as metadata. When an end-user triggers the execution of Apex, perhaps by clicking a button or accessing a Visual force page, the platform application server retrieves the compiled instructions from the metadata and sends them through the runtime interpreter before returning the result. The end-user observes no differences in execution time from standard platform requests.

## **VISUALFORCE:**

Visualforce is a framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Force.com platform. The Visualforce framework includes a tag-based markup language, similar to HTML. In the Visualforce markup language, each Visualforce tag corresponds to a coarse or fine-grained user interface component, such as a section of a page, a related list, or a field. The behavior of Visualforce components can either be controlled by the same logic that is used in standard Salesforce pages, or developers can associate their own logic with a controller class written in Apex.

Force.com platform offers separate tools for defining:

- The structure of the data—that is, the *data model*
- The rules that detail how that data can be manipulated—that is, the *business logic*
- The layouts that specify how that data should be displayed—that is, the *user interface*

A visualforce page definition consists of two primary elements:

- Visualforce markup
- A Visualforce controller

### **Visualforce Markup**

Visualforce markup consists of Visualforce tags, HTML, JavaScript, or any other Web-enabled code embedded within a single `<apex:page>` tag. The markup defines the user interface components that should be included on the page, and the way they should appear.

### **Visualforce Controllers**

A Visualforce controller is a set of instructions that specify what happens when a user interacts with the components specified in associated Visualforce markup, such as when a user clicks a button or link. Controllers also provide access to the data that should be displayed in a page, and can modify component behavior. A developer can either use a standard controller provided by the Force.com platform, or add custom controller logic with a class written in Apex.

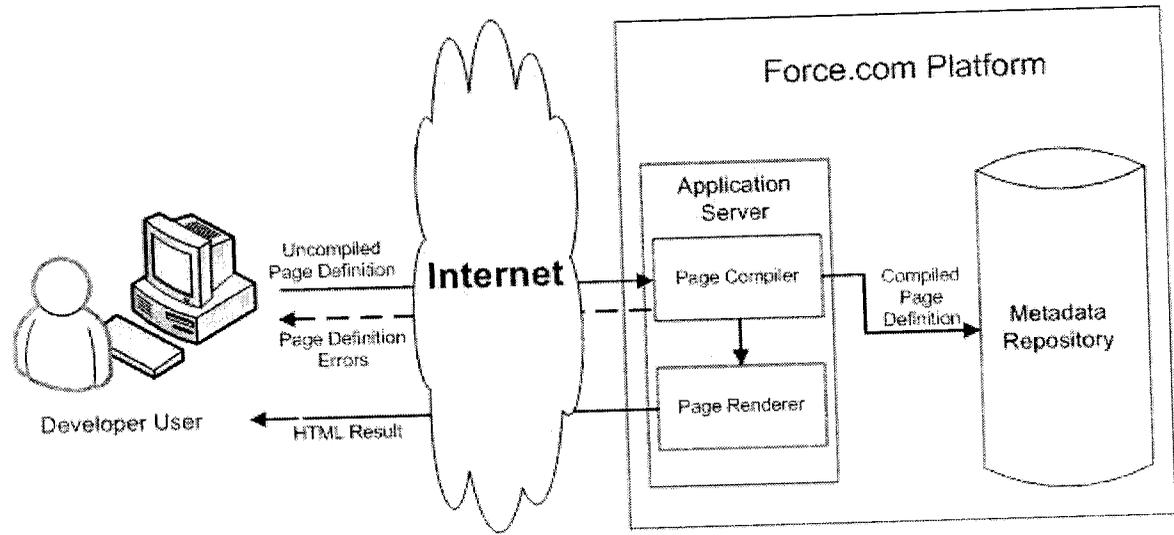
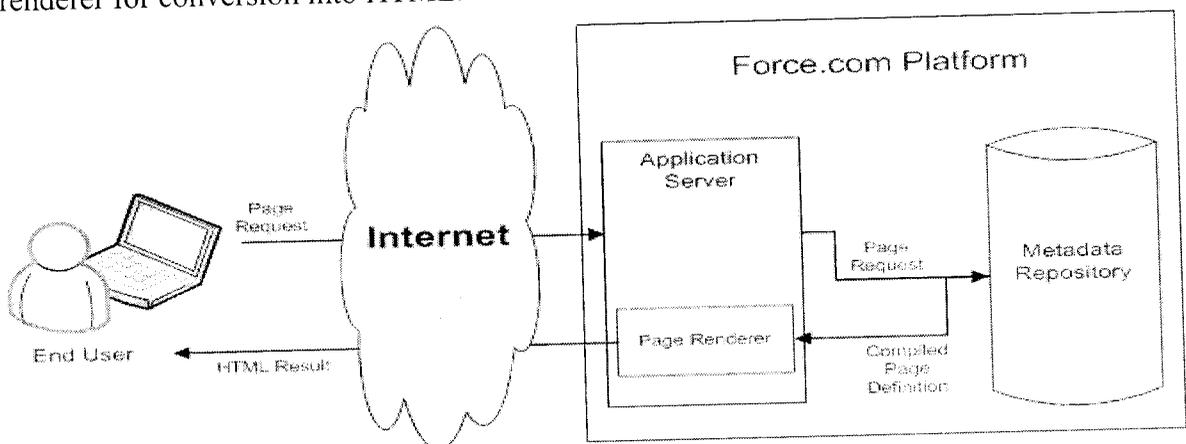


Figure 1.6: Visualforce System Architecture - Development Mode

When a developer finishes writing a Visualforce page and saves it to the platform, the platform application server attempts to compile the markup into an abstract set of instructions that can be understood by the Visualforce renderer. If compilation generates errors, the save is aborted and the errors are returned to the developer. Otherwise, the instructions are saved to the metadata repository and sent to the Visualforce renderer. The renderer turns the instructions into HTML and then refreshes the developer's view, thereby providing instantaneous feedback to the developer for whatever changes were made in the markup. The architecture diagram below shows the process flow when a non-developer user requests a Visualforce page. Because the page is already compiled into instructions, the application server simply retrieves the page from the metadata repository and sends it to the Visualforce renderer for conversion into HTML.



## CHAPTER 5

### PROJECT DESCRIPTION

#### 5.1 PROBLEM DEFINITION:

An organization may contain many branches which are located at different places which may use of own interested database to store the information. The data available is huge and not in unique format. It is difficult for a user to manually analyze these data and make decisions. The problem here is to use the cloud computing model for storing all the informations so that it can be accessed anytime and anywhere via the software which can be accessed as a service on subscription or it will be of pay per use model. A system must be designed in a device independent manner which provides a user interactive interface to access the software and the data elsewhere in the world . An effective software as a service must be designed for organization information tracking based on the required functionality on a on-demand basis.

#### 5.2 PROJECT OVERVIEW:

In this project, software as a service takes place ,where project codes are compiled and stored in the force.com platform as metadata. The source for the project is the force.com platform and the infrastructure provided by them. The entire project is divided in to two phases as follows:-

- **Phase I:** Web service analysis and the cloud platform study.
- **Phase II:** software is developed on the remote force.com platform.

#### 5.3 MODULES:

##### TRACKME SYSTEM:

The system is composed of a number of sub-modules, which correspond to each stage of development. These stages are A) Developing the custom objects B) Choosing the relationship between the custom objects C) Developing required validation rules D) Customizing using the visualforce controllers and pages. E) Deploying it into the production

**5.3.1. User Login:**User should log in to use the system.The user must be the authorised one, for checking whether the person is authorised , username and password is checking,if its correct , the user can use the system as a service, suppose if its wrong it will error message like username or password incorrect, please enter correct username and password.

### **5.3.2 Track Me Search (Drill Down and Drill Up):**

Drill down and drilling up of the results can be done by querying as required. Queries are one of the things that make databases so powerful. A "query" refers to the action of retrieving data from your database. Usually, you will be selective with how much data you want returned. If you have a lot of data in your database, you probably don't want to see everything. More likely, you'll only want to see data that fits a certain criteria.For example, you might only want to see how many users in your database live in a given city. Or you might only want to see which individuals have registered with your database within a given time period. Database queries consist of patterns which match either individual segments, sequences of segments or parts of the hierarchical description of an utterance. The result of a query is a segment list which contains the labels and start and end times of each segment or sequence of segments which matched the query, along with the utterances to which they belong.

Queries can relate to any level of the hierarchy defined by the template file.The type of segments returned depends on the form of the query.There are two basic kinds of query: simple queries which relate to single segments, and compound queries which relate to either sequences of segments or to segments at different levels in the hierarchy.

### **5.3.3 Machine Maintenance History**

This module is developed for maintaining the machine maintenance records which have relationship with other custom and standard objects as well. New data can be entered based on the relationship to parent if any. Data can also be deleted which also deletes the parent record based on the relationship defined.

### **5.3.4 Reports**

Reports are defined on various criteria. The reports are based on the record count, number of cases closed as of date, number of opened cases, aging of the cases, location of the cases and other reports as well.

## CHAPTER 6

### RESULTS AND DISCUSSION

For experimental study sample records have been created and these are accessed independently on all other systems. This is done mainly using the activation link option provided by the force.com platform for accessing the same on other systems i.e., if we are accessing the system from untrusted network the system will ask for authorization which will send a email with the activation link through which we can access the system. This can be done dynamically by providing the trusted network IP ranges. In future it will be developed into a package so that it can be integrated with the other systems.

	Software	Software as a Service
Instant software updates	No	Yes
Improved Performance	No	Yes
Unlimited Storage	No	Yes
Increased Data Reliability	No	Yes
Device dependent	Yes	No
Latest version available	No	Yes
Universal document access	No	Yes
Does not require internet connection	Yes	No

Table 1. Comparative study

## **CHAPTER 7**

### **CONCLUSION AND FUTURE OUTLOOK**

Thus a new software as a service have been developed using the force.com platform. This software is device independent and can be accessed anywhere and at anytime. It is provided with appropriate custom fields and validation rules to provide more functionality. The software is also tested using the test codes and this follows the international coding standards which will not make the platform hang at any instance. This is developed into a managed package which can be installed in force.com platform more than once by getting the license, but a single instance of the system will be executing at backend which upon on more requests it will be virtualized and allocated. In future the system will be developed into a downloadable package which will helps to integrate with other applications.

## APPENDIX I

### SOURCE CODE

#### Coding for Accessing TrackMe Search:

##### a) Source code for TrackMe Search:

```

public class TrackMeSearch_Controller {
public List<Machine_Barcode__c> MachineBR {get;set;}
public List<SelectOption> StateName;
public List<SelectOption> citiesName;
public List<SelectOption> streetsName;
public List<SelectOption> barcodeNumber;
public List<SelectOption> SubIdNumber;
String SearchMachineBarcode;
String Str = '%';

public String getSearchMachineBarcode() {
    return this.SearchMachineBarcode;
}

public void setSearchMachineBarcode(String search) {
    this.SearchMachineBarcode = search;
}

public List<SelectOption> getstates()
{
    try
    {
        if (StateName==null){
            StateName= new List<SelectOption>();
            String test=System.Label.State;
            System.debug(test);
            StateName.add(new SelectOption(", '- Select State -'"));

            for (SObject loc : [Select State__c, count(name) from Location_Master__c group
by State__c]) {
                SelectOption state = new SelectOption((String)loc.get('State__c'),
                (String)loc.get('State__c'));
                StateName.add(state);
            }
        }
    }
}Catch(Exception e){
    System.debug(e);
}

```

```
public String selectedState {get; set;}
```

```
public List<SelectOption> getcities()
{
```

```
    try
    {
        citiesName= new List<SelectOption>();
        String test1=System.Label.City;
        System.debug(test1);
        citiesName.add(new SelectOption(", '- Select City -'"));
        for (SObject loc : [Select City__c, count(name) from Location_Master__c
            where State__c = :selectedState group by City__c]) {
            SelectOption city = new SelectOption((String)loc.get('City__c'),
                (String)loc.get('City__c'));
            citiesName.add(city);
            System.debug(city);
        }
        System.debug('Cities size: '+ citiesName.size());
    } Catch(Exception e){
        System.debug(e);
    }
    return citiesName;
}
```

```
public String selectedCity {get; set;}
```

```
public List<SelectOption> getstreets()
{
```

```
    try
    {
        streetsName= new List<SelectOption>();
        String test2=System.Label.Address;
        System.debug(test2);
        streetsName.add(new SelectOption(", '- Select Location -'"));
        for (Location_Master__c loc : [Select Id, name, Address_Line_1__c,
            Address_Line_2__c, Zip_Code__c from Location_Master__c
            where State__c = :selectedState and City__c = :selectedCity]) {
            String label = loc.Address_Line_1__c;
            if (loc.Address_Line_2__c != null) {
                label = label+' , '+loc.Address_Line_2__c;
            }
            SelectOption street = new SelectOption(label, label);
            streetsName.add(street);
        }
    } Catch(Exception e){
        System.debug(e);
    }
    return streetsName;
}
```

```

public String selectedStreet {get; set;}

public List<SelectOption> getbarcodeNum()
{
    try
    {
        barcodeNumber= new List<SelectOption>();
        barcodeNumber.add(new SelectOption("-", "Select Barcode -"));
        for (sObject mac : [Select Barcode__c, count(name) from Machine_Barcode__c
            where State__c like :selectedState and City__c like :selectedCity
            and Address__c like :selectedStreet group by Barcode__c]) {
            SelectOption barcode = new SelectOption((String)mac.get('Barcode__c'),
                (String)mac.get('Barcode__c'));
            barcodeNumber.add(barcode);
        }
    } Catch(Exception e){
        System.debug(e);
    }
    return barcodeNumber;
}

public String selectedBarcodeNum {get; set;}

public List<SelectOption> getSubIdNum()
{
    try
    {
        SubIdNumber= new List<SelectOption>();
        SubIdNumber.add(new SelectOption("-", "Select SubId -"));
        for (Machine_Barcode__c sb : [Select Id, Sub_Id__c, name, Barcode__c from
            Machine_Barcode__c where State__c like :selectedState and City__c like
            :selectedCity and Address__c like :selectedStreet and Barcode__c like
            :selectedBarcodeNum]) {
            String label = sb.Sub_Id__c;

            SelectOption SubId = new SelectOption(label, label);
            SubIdNumber.add(SubId);
        }
    } Catch(Exception e){
        System.debug(e);
    }
    return SubIdNumber;
}

```

```

public Id barId;
public PageReference searchLocation(){

```

```

    Integer flag=0;
    str='%'+SearchMachineBarcode+'%';
    PageReference pageRef = null ;

```

```

try

```

```

{

```

```

    String State='%';
    String City='%';
    String Street='%';
    String BarcodeNumber='%';
    String SubIdNumber='%';

```

```

if(selectedState!=null)
    State=selectedState;
if(selectedCity!=null)
    City=selectedCity;
if(selectedStreet!=null)
    Street=selectedStreet;
if(selectedBarcodeNum!=null)
    BarcodeNumber=selectedBarcodeNum;
if(selectedSubIdNum!=null)
    SubIdNumber=selectedSubIdNum;

```

```

System.Debug('*****State:'+selectedState+'*****
*****City:'+selectedCity+'*****
****Address:'+selectedStreet);

```

```

    List<Machine_Barcode__c> MachineBR1= [Select id, Sub_Id__c, barcode__c,
name,State__c,City__c,Address__c from Machine_Barcode__c];

```

```

    System.Debug('*****BarcodeName*****'+MachineBR1);

```

```

for(Machine_Barcode__c m:MachineBR1){

```

```

    if(m.State__c==selectedState && m.City__c==selectedCity &&
m.Address__c==selectedStreet && m.Barcode__c==selectedBarcodeNum &&
m.Sub_Id__c==selectedSubIdNum){

```

```

        barId=m.Id;
        flag++;

```

```

        pageRef = new PageReference('/apex/MachineBarcodePage?id='+barId);
        pageRef.setRedirect(true);

```

```

    }

```

```

if(flag==0){
  if(SearchMachineBarcode==""){
    MachineBR=[select id,name,Address__c,Barcode__c,City__c,Location_Master__c,
      Location_Description__c,Machine_Description__c,
      Machine_Master__c,State__c,Sub_Id__c,Zipcode__c from
      Machine_Barcode__c where State__c like :State
      and City__c like :City and Address__c like :Street and barcode__c
      like :BarcodeNumber and Sub_Id__c like :SubIdNumber];
  }
  else{
    MachineBR=[select id,name,Address__c,Barcode__c,City__c,Location_Master__c,
      Location_Description__c,Machine_Description__c,
      Machine_Master__c,State__c,Sub_Id__c,Zipcode__c from
      Machine_Barcode__c where Barcode__c like :str or name like :str];
  }
}
}
}Catch(Exception e){
System.debug(e);
}

return pageRef;
}

```

## b)Source code for Testing TrackMe Search:

```

Public Static TestMethod void TestTrackMeSearch_Controller(){
  Location_Master__c LM = new Location_Master__c(Address_Line_1__c=
'Foot Hill Commons',Address_Line_2__c='200, 25 th ave west',City__c='Dallas',
State__c='TX',Zip_Code__c='75081');
insert LM;

  Machine_Master__c MC = new Machine_Master__c(Description__c='cooling
system');
insert MC;

  Machine_Barcode__c MB = new
Machine_Barcode__c(Barcode__c='123456765',Sub_Id__c='34563216',Location_Ma
ster__c=LM.Id,Machine_Master__c=MC.Id);
insert MB;
Update MB;

PageReference PRef = Page.TrackmeSearch;
Test.setCurrentPageReference(PRef);

```

```
ApexPages.currentPage().getParameters().put('id',MB.id);
```

```
trc.getSearchMachineBarcode();
trc.setSearchMachineBarcode('name');
trc.searchLocation();
trc.barId=MB.Id;
trc.getstates();
trc.getcities();
trc.getstreets();
trc.getbarcodenum();
trc.getsubidnum();
```

```
trc = new TrackMeSearch_Controller();
trc.SearchMachineBarcode="";
trc.selectedState='TX';
trc.selectedcity='Dallas';
trc.selectedstreet='Foot Hill Commons , 200, 25 th ave west';
trc.selectedBarcodeNum='123456765';
trc.selectedSubIdNum='34563216';
```

```
String nextPage = trc.searchLocation().getUrl();
nextPage = trc.searchLocation().getUrl();
```

```
}
}
```

## b) Source code for Search:

```
public class TrackMeSearch_ControllerOut {
```

```
public Machine_Barcode__c barcode{get;set;} 
```

```
Public List<Machine_Maintenance__c> RList = new List<Machine_Maintenance__c>();
```

```
Public List<Order__c> CaseList = new List<Order__c>();
```

```
Public List<Machine_Barcode__c> Mbarcode = new List<machine_barcode__c>();
```

```
Public List<Machine_Detail_Record__c> RelatedMaintenance = new
```

```
List<Machine_Detail_Record__c>();
```

```
Public List<Attachment> Attached = new List<Attachment>();
```

```
Public List<Note> Notes = new List<Note>();
```

```
public TrackMeSearch_ControllerOut(){
```

```
try
```

```
{
barcode=[select id,name,Address__c,Barcode__c,City__c,Location_Master__r.Name,
```

```
Location_Description__c,Machine_Description__c,Machine_Master__r.Name,
State__c,Sub_Id__c,Zipcode__c,CreateDate,CreatedById,CreatedBy.Name
```

```

}Catch(Exception e){
System.Debug(e);
}
}

```

```

Public List<Machine_Barcode__c> getMbarcode(){
    try
    {

```

```

        Mbarcode = [select
Name,Id,Barcode__c,Expiration_date__c,Sub_Id__c,Address__c,city__c,state__c,zipcode__
c from Machine_Barcode__c where machine_master__c=:barcode.Machine_master__c
ORDER BY zipcode__c ASC];
    }Catch(Exception e){
System.Debug(e);
}
    return Mbarcode;
}

```

```

Public List<Machine_Detail_Record__c> getRelatedMaintenance(){
    try
    {

```

```

        RelatedMaintenance=[Select Name,Id,Attribute__c,Machine_Master__r.Name,value__c
from Machine_Detail_Record__c where Machine_Master__c=:barcode.Machine_Master__c
Order BY Name ASC];
    }Catch(Exception e){
System.Debug(e);
}
    return RelatedMaintenance;
}

```

```

Public List<Machine_Maintenance__c> getRList(){
    try
    {

```

```

        RList = [Select id, Name,
On_Off_Indicator__c,Address__c,City__c,Description__c,Fire_Order_Creation__c,
Last_Case_Complete_Date__c,Last_case_creation_Date__c,Location_Description__c,Locati
on_Master__c,
Machine_Barcode__c,Machine_Description__c,Machine_Master__r.Name,Maintenance_Da
ys__c,Next_Scheduled_Maintenance__c,
Part_Description__c,Parts_Master__r.Name,State__c,CreatedBy.Name,CreatedById.Created
Date,Zip_Code__c from Machine_Maintenance__c where
Machine_Barcode__c=:barcode.Id Order by Name ASC];
    }
}

```

```

List<Order__c> cases = [select
Name,Id,Description__c,Priority__r.Name,Close_Date__c,CreateDate,CreatedBy.Name,Du
e_Date__c,Old_Priority__c,Current_Assigned_Department__r.Name,Old_Status__c,Status_n
ew__c from Order__c where Machine_Maintenance__c = :MM.Id ORDER BY Name
ASC];
System.Debug('Total Cases are #####'+Cases);

for(Order__c CL:cases){
CaseList.add(CL);
}

}

}Catch(Exception e){
System.Debug(e);
}
return RList;
}

Public List<Order__c> getCaseList(){
return CaseList;
}

Public Class AttachClass{

public String LmDate {get;set;}
public String cDate {get;set;}
public String AttachName {get;set;}
public String AttachId {get;set;}

Public AttachClass(String LastD, String Date1, String Name1, string AttachId1){

AttachId=AttachId1;
LmDate=LastD;
cDate=Date1;
AttachName=Name1;
}
}

Public List<AttachClass> ListAttach=new List<AttachClass>();

Public List<AttachClass> getListAttach(){
ListAttach.clear();
List<Attachment> attach = [select Id, Name, LastModifiedDate, CreateDate, Body,
Description from Attachment where ParentId=:barcode.Machine_Master__c];

```

```

AttachClass wrapperClass = new AttachClass(a.LastModifiedDate.format('MM/d/yyyy,
h:mm a'),a.CreatedDate.format('MM/d/yyyy, h:mm a'),a.Name,a.Id);
ListAttach.add(wrapperClass);

```

```

}
return ListAttach;
}

```

```

Public Class NoteClass{

```

```

public String LDate {get;set;}
public String cdDate {get;set;}
public String NoteName {get;set;}
public String NoteId {get;set;}

```

```

Public NoteClass(String LastDate, String cdDate1, String NoteName1, string NoteId1){

```

```

NoteId = NoteId1;
LDate = LastDate;
cdDate = cdDate1;
NoteName = NoteName1;
}
}

```

```

Public List<NoteClass> ListNote=new List<NoteClass>();

```

```

Public List<NoteClass> getListNote(){

```

```

ListNote.clear();
List<Note> Notes = [select Id, Title, LastModifiedDate, CreatedDate, Body from Note
where ParentId=:barcode.Machine_Master__c];

```

```

for(Note n:notes){
NoteClass Nclass = new NoteClass(n.LastModifiedDate.format('MM/d/yyyy, h:mm
a'),n.CreatedDate.format('MM/d/yyyy, h:mm a'),n.Title,n.Id);
ListNote.add(Nclass);

```

```

}
return ListNote;
}

```

```

Public Static TestMethod void TestTrackMeSearch_ControllerOut(){

```

```

Location_Master__c LM = new Location_Master__c(Address_Line_1__c='Foot Hill
Commons',Address_Line_2__c='200, 25 th ave
west',City__c='Dallas',State__c='TX',Zip_Code__c='75081');

```

```

Machine_Master__c MC = new Machine_Master__c(Description__c='cooling
system');
Insert MC;

```

```

Machine_Barcode__c MB = new
Machine_Barcode__c(Barcode__c='123456765',Sub_Id__c='34563216',Location_Master__c
=LM.Id,Machine_Master__c=MC.Id);
insert MB;
Update MB;

```

```

Machine_Maintenance__c MM = new
Machine_Maintenance__c(Machine_Barcode__c=MB.Id,Maintenance_Days__c=1,Location
_Master__c=LM.Id,Machine_Master__c=MC.Id);

```

```

TrackMeSearch_ControllerOut controller = new TrackMeSearch_ControllerOut();

```

```

PageReference Pageref = Page.MachineBarcodePage;
Test.setCurrentPageReference(Pageref);

```

```

ApexPages.currentPage().getParameters().put('id',MB.id);

```

```

List<Machine_Maintenance__c> RList = new List<Machine_Maintenance__c>();
List<Order__c> CaseList = new List<Order__c>();
List<Machine_Barcode__c> Mbarcode = new List<Machine_Barcode__c>();
List<Machine_Detail_Record__c> RelatedMaintenance = new
List<Machine_Detail_Record__c>();

```

```

RList=controller.getRlist();
CaseList=controller.getCaseList();
Mbarcode=controller.getMbarcode();
RelatedMaintenance=controller.getRelatedMaintenance();

```

```

}

```

### c) Visual force Source code for TrackMe Search:

```

public abstract class TrackMeCaseAssignedTeam {
private Order_Assignment__c currentDeptAssignment;
public String currentUser {get; private set;}
private boolean isPM;
private boolean isTL;
private boolean isTM;
public boolean isTrackMeAdmin {get; private set;}
public boolean isSystemAdministrator {get; private set;}

public TrackMeCaseAssignedTeam (Id orderId) {

```

```

protected void init(Id orderId) {
    currentUser = UserInfo.getUserId();

    if (orderId!=null) {
        Order_Assignment__c[] oas = [select Id, Department__c,
Department__r.Property_Manager__c, Department__r.Primary_User__c,
Department__r.Secondary_User__c from Order_Assignment__c
        where Order__c = :orderId
        and Expired__c = 'NO'
        and Level__c = 'Department'];
        if (oas.size() > 0) {
            currentDeptAssignment = oas[0];
        }
    }

    if (currentDeptAssignment != null) {
        if (currentDeptAssignment.Department__r.Primary_User__c == currentUser ||
currentDeptAssignment.Department__r.Secondary_User__c == currentUser) {
            if (currentDeptAssignment.Department__r.Property_Manager__c) {
                isPM = true;
                isTL = true;
                isTM = true;
                System.debug('PM, TL and TM');
            } else {
                isPM = false;
                isTL = true;
                isTM = true;
                System.debug('TL and TM');
            }
        } else if ([select Id from Department_User__c
        where Department__c = :currentDeptAssignment.Department__c
        and User__c = :currentUser
        and (Expiration_Date__c = null or Expiration_Date__c > :System.now())].size()
>0) {
            isPM = false;
            isTL = false;
            isTM = true;
            System.debug('TM');
        } else {
            isPM = false;
            isTL = false;
            isTM = false;
            System.debug('no membership');
        }
    } else {
        isPM = true;
        isTL = true;
    }
}

```

```

Profile profile = [Select Id, Name from Profile where Id = :UserInfo.getProfileId()];
if (profile.Name == 'System Administrator') {
    isSystemAdministrator = true;
    isTrackMeAdmin = false;
} else if (profile.Name == 'TrackMe Administrator') {
    isSystemAdministrator = false;
    isTrackMeAdmin = true;
} else {
    isSystemAdministrator = false;
    isTrackMeAdmin = false;
}
}

```

```

public boolean getIsPropertyManager() {
    return isPM;
}

```

```

public boolean getIsTeamLead() {
    return isTL;
}

```

```

public boolean getIsTeamMember() {
    return isTM;
}
}

```

#### d) Visual force Source code for TrackMe Search:

```

<apex:page controller="TrackMeSearch_Controller">
    <apex:form >
        <apex:pageBlock title="Track Me Search" mode="edit">
            <apex:pageBlockSection title="Enter the Following Details" columns="3">
                <apex:pageblockSectionItem >
                    <apex:outputText value=""/>
                </apex:pageblockSectionItem>
                <apex:pageblockSectionItem >
                    <apex:outputText value=""/>
                </apex:pageblockSectionItem>
                <apex:pageblockSectionItem >
                    <apex:outputText value="State : "/>
                    <apex:selectList value="{!selectedState}" id="locState" size="1">
                        <apex:selectOptions value="{!states}"></apex:selectOptions>
                    <apex:actionSupport event="onchange" reRender="locCity, locations,

```

```

</apex:pageblockSectionItem>

<apex:pageblockSectionItem >
  <apex:outputText value=""/>
</apex:pageblockSectionItem>

<apex:pageblockSectionItem >
  <apex:outputText value=""/>
</apex:pageblockSectionItem>

<apex:pageblockSectionItem >
  <apex:outputText value="City : "/>
  <apex:selectList value="{!selectedCity}" id="locCity" size="1">
    <apex:selectOptions value="{!cities}"></apex:selectOptions>
    <apex:actionSupport event="onchange" reRender="locations,
barcode, subId"/>
  </apex:selectList>
</apex:pageblockSectionItem>

<apex:pageblockSectionItem >
  <apex:outputText value="Search Name/Barcode:"/>
  <apex:inputText value="{!SearchMachineBarcode}"/>
</apex:pageblockSectionItem>

<apex:pageblockSectionItem >
  <apex:outputText value=""/>
</apex:pageblockSectionItem>

<apex:pageblockSectionItem >
  <apex:outputText value="Address : "/>
  <apex:selectList value="{!selectedStreet}" id="locations" size="1">
    <apex:selectOptions value="{!streets}"></apex:selectOptions>
    <apex:actionSupport event="onchange" reRender="barcode, subId"/>
  </apex:selectList>
</apex:pageblockSectionItem>

<apex:pageblockSectionItem >
  <apex:outputText value=""/>
</apex:pageblockSectionItem>

<apex:pageblockSectionItem >
  <apex:outputText value=""/>
</apex:pageblockSectionItem>

<apex:pageblockSectionItem >
  <apex:outputText value="Barcode : "/>
  <apex:selectList value="{!selectedBarcodeNum}" id="barcode" size="1">
    <apex:selectOptions value="{!barcodeNum}"></apex:selectOptions>
    <apex:actionSupport event="onchange" reRender="subId"/>
  </apex:selectList>
</apex:pageblockSectionItem>

```

```
</apex:pageblockSectionItem>
```

```
<apex:pageblockSectionItem >
  <apex:outputText value=""/>
</apex:pageblockSectionItem>
```

```
<apex:pageblockSectionItem >
  <apex:outputText value=""/>
</apex:pageblockSectionItem>
```

```
<apex:pageblockSectionItem >
  <apex:outputText value="SubId : "/>
  <apex:selectList value="{!selectedSubIdNum}" id="subId" size="1">
    <apex:selectOptions value="{!SubIdNum}"></apex:selectOptions>
  </apex:selectList>
</apex:pageblockSectionItem>
```

```
</apex:pageBlockSection>
```

```
<center><apex:commandButton value="Search" action="{!searchLocation}"
reRender="MachineBRList"/></center>
```

```
</apex:pageBlock>
```

```
<apex:pageBlock title="Equipment Details" rendered="true">
```

```
<apex:pageBlockTable id="MachineBRList" value="{!MachineBR}" var="brc">
  <apex:column >
```

```
<apex:facet name="header">
```

```
<apex:outputText value="Barcode"/>
```

```
</apex:facet>
```

```
<apex:outputLink
```

```
value="/apex/MachineBarcodePage?id={!brc.id}">{!brc.Barcode__c}</apex:outputLink>
```

```
</apex:column>
```

```
<apex:column >
```

```
<apex:facet name="header">
```

```
<apex:outputText value="SubId"/>
```

```
</apex:facet>
```

```
<apex:outputLink
```

```
value="/apex/MachineBarcodePage?id={!brc.id}">{!brc.Sub_Id__c}</apex:outputLink>
```

```
</apex:column>
```

```
<apex:column >
```

```
<apex:facet name="header">
```

```
<apex:outputText value="Machine Description"/>
```

```
</apex:facet>
```

```
<apex:outputLink
```

```
value="/apex/MachineBarcodePage?id={!brc.id}">{!brc.Machine_Description__c}</apex:outputLink>
```

```
</apex:column>
```

```

    <apex:facet name="header">
      <apex:outputText value="Address"/>
    </apex:facet>
    <apex:outputLink
value="/apex/MachineBarcodePage?id={!brc.id}">{!brc.State__c}, {!brc.City__c},
{!brc.Address__c}, {!brc.Zipcode__c}</apex:outputLink>
    </apex:column>
  </apex:pageBlockTable>
</apex:pageBlock>

</apex:form>
</apex:page>

<apex:page showHeader="false" standardController="Order__c" tabStyle="Order__c"
extensions="TrackMeSiteHomePageControllerExtension,TrackMeCaseControllerExtension"
>
  <style>
    .Outage { color: red; font-weight: strong;}
    .Regular { color: green; font-weight: strong;}
    .topAlign {height='100%';vertical-align:text-top;align: left}
  </style>
  <apex:composition template="SiteTemplate">
    <apex:define name="body">
      <!-- Scrolling announcements by location -->
      <apex:pageBlock title="Announcements">
        <TABLE cellSpacing="0" cellPadding="0" width="100%" border="1">
          <TBODY>
            <apex:repeat value="{!announcements}" var="annEntry"
id="repeatAnnouncements">
              <TR>
                <TD vAlign="top" width="30%">{!annEntry.location}</TD>
                <TD vAlign="top" width="70%">
                  <marquee behavior="scroll" direction="left">
                    <apex:repeat value="{!annEntry.announcements}" var="ann">
                      <apex:outputText styleClass="{!ann.Type__c}"
value="{!ann.Message__c}"></apex:outputText>
                      &nbsp;
                    </apex:repeat>
                  </marquee>
                </TD>
              </TR>
            </apex:repeat>
          </TBODY>
        </TABLE>
      </apex:pageBlock>
      <!-- End Scrolling announcements by location -->

      <apex:panelGrid columns="2" columnClasses="topAlign">

```

```

    <apex:sectionHeader title="Menu" subtitle="Authorized Users"/>
    <!--apex:outputLink value="/home/home.jsp">TrackMe
Home</apex:outputLink-->
    <br/>
    <apex:outputLink
value="{!URLFOR($Action.Order__c.Tab,$ObjectType.Order__c)}">Manage
Cases</apex:outputLink>
</apex:panelgroup>
<apex:panelgroup >
    <apex:sectionHeader title="TrackMe" subtitle="Case Tracking"/>
    Welcome to TrackMe application. Please fill the form below to submit a Case.
    <apex:form id="editPage">
    <apex:pageBlock title="Create A Case" mode="edit">
        <apex:pageMessages />
        <apex:pageBlockButtons >
            <apex:commandButton action="{!quicksave}" value="Save"
rendered="{!Order__c.Id == null}"/>
            <apex:commandButton value="New"
onclick="window.top.location='{!$Site.Prefix}'; return false;" rendered="{!Order__c.Id !=
null}"/>
        </apex:pageBlockButtons>
        <apex:pageBlockSection title="Location Information" columns="1">
            <c:LookupComponent value="{!Order__c.Location__c}"
object="Location_Master__c"
            filterFields="Name,Address_Line_1__c, City__c, State__c,
Zip_Code__c" sectionTitle="Select Location"
            rowsPerPage="2">
                </c:LookupComponent>
            </apex:actionRegion>
            <c:LocationLookup value="{!Order__c.Location__c}"></c:LocationLookup>
            <apex:inputField value="{!Order__c.Floor_Suite__c}"/>
        </apex:pageBlockSection>
        <apex:pageBlockSection title="Case Information" columns="1">
            <apex:outputField value="{!Order__c.Name}" rendered="{!Order__c.Id !=
null}"/>
            <apex:pageBlockSectionItem >
                <apex:outputLabel value="Request Type" for="requestTypeList"/>
                <apex:selectList value="{!Order__c.Request_Type__c}"
id="requestTypeList" size="1">
                    <apex:selectOptions value="{!requestTypes}"/>
                </apex:selectList>
            </apex:pageBlockSectionItem>
            <apex:inputField value="{!Order__c.Description__c}" style="width: 60%"/>
        </apex:pageBlockSection>
        <apex:pageBlockSection title="Tenant Information" columns="2">
            <apex:inputField value="{!Order__c.Reporting_Tenant_Name__c}"/>
            <apex:inputField value="{!Order__c.Reporting_Tenant_Phone__c}"/>
            <apex:inputField value="{!Order__c.Reporting_Tenant_Email__c}"/>
            <apex:inputField value="{!Order__c.Best_Time_To_Contact__c}"/>

```

```
        </apex:pageBlock>  
        </apex:form>  
    </apex:panelgroup>  
</apex:panelGrid>  
</apex:define>  
</apex:composition>  
  
<apex:page>
```

## APPENDIX II

## SNAP SHOTS

### TRACK ME:

**TrackMe Search**

Enter the Following Details

Search Name/Barcode:

State:

City:

Address:

Barcode:

Subid:

**Equipment Details**

Barcode	Subid	Machine Description	Address
---------	-------	---------------------	---------

### TRACK ME - EQUIPMENT DETAILS:

**TrackMe Search**

Enter the Following Details

Search Name/Barcode:

State:

City:

Address:

Barcode:

Subid:

**Equipment Details**

Barcode	Subid	Machine Description	Address
87678696	8576876697	GE Dish Washer	IL, Champion Ed Tower, 825 Kenwood Road, 61825
4546546	787696	HVAC - Carrier	IL, Chicago, 1300 Winstone ST., 61822
5438112345	010101010101	cooling system	IL, Chicago, 1300 Winstone ST., 61822
23423456	676876878	cooling system	WA, Lynnwood, Foot Hill Commons, 123 NE 8th Street, 98067
4567567	788787	cooling system	WA, Lynnwood, Foot Hill Commons, 123 NE 8th Street, 98067
4568	8876	water filter	WA, Lynnwood, Foot Hill Commons, 123 NE 8th Street, 98067
1234567890	304-7-003	HVAC - Carrier	WA, Lynnwood, Foot Hill Commons, 123 NE 8th Street, 98067

### TRACK ME USING BARCODE INFORMATION:

**TrackMe Search**

Enter the Following Details

Search Name/Barcode:

State:

City:

Address:

Barcode:

Subid:

**Equipment Details**

Barcode	Subid	Machine Description	Address
87678696	8576876697	GE Dish Washer	IL, Champion Ed Tower, 825 Kenwood Road, 61825
4546546	787696	HVAC - Carrier	IL, Chicago, 1300 Winstone ST., 61822
5438112345	010101010101	cooling system	IL, Chicago, 1300 Winstone ST., 61822
23423456	676876878	cooling system	WA, Lynnwood, Foot Hill Commons, 123 NE 8th Street, 98067
4567567	788787	cooling system	WA, Lynnwood, Foot Hill Commons, 123 NE 8th Street, 98067
4568	8876	water filter	WA, Lynnwood, Foot Hill Commons, 123 NE 8th Street, 98067
1234567890	304-7-003	HVAC - Carrier	WA, Lynnwood, Foot Hill Commons, 123 NE 8th Street, 98067

## MACHINE INFORMATION WITH DEPLOYED LOCATIONS:

The screenshot displays the 'Machine Barcode' page in the TrackMe application. The top navigation bar includes Home, Departments, TrackMe Search, TrackMe Profiles, Status, Dashboards, Reports, Screens, Request Types, SLAs, Business Flows, Cases, TrackMe Users, Equipments, Parts, and Locations. The left sidebar contains 'Recent Items' with links for Finance, Complete, Account Corrected, and several Q-00232 through Q-00234 items. The main content area is titled 'Machine Barcode' and contains two sections: 'Information' and 'Location Information'. The 'Information' section shows Barcode: 4545546, Subid: 787895, Machine Master: M0001, Machine Description: HVAC - Carrier, and Created By: Subha Admin, 8/9/2010 9:28 AM. The 'Location Information' section shows Location: L-0005, Address: 1300 Wynstone ST., City: Chicago, State: IL, and Zipcode: 61822. Below this is a table with tabs for Machine Maintenances, Cases, Deployed Locations, Related Machine Details, Attachments, and Notes. The 'Deployed Locations' tab is active, showing a table with columns Barcode, Subid, Address, and Expiration Date. The table contains three rows of data.

## TRACK ME- WITH RELATED MACHINE DETAILS:

This screenshot shows the 'Machine Master' page in the TrackMe application. The navigation and sidebar are identical to the previous screenshot. The main content area is titled 'Machine Master' and shows Barcode: 4545546, Subid: 787895, Machine Description: HVAC - Carrier, and Created By: Subha Admin, 8/9/2010 9:28 AM. Below this is the 'Location Information' section with the same details as the previous screenshot. At the bottom, there is a table with tabs for Machine Maintenances, Cases, Deployed Locations, Related Machine Details, Attachments, and Notes. The 'Related Machine Details' tab is active, displaying a table with columns Machine Master, Attribute, and Value. The table shows Machine Master: M0001, Attribute: A/C, and Value: model#123.

## TRACK ME- STANDARD PAGE FOR MACHINE MASTER :

The screenshot shows the 'Note' page for machine master M0001 in the TrackMe application. The navigation bar and sidebar are consistent. The main content area is titled 'Note M0001' and includes a 'Back to List: Custom Object Definitions' link. Below this is the 'Note Detail' section with 'Edit' and 'Delete' buttons. The note details are: Note Owner: Mohan Elango, Related To: M0001, Private: (checked), Title: Test, Body: Test the machine details custom Visualforce Page., and Created By: Mohan Elango, 1/19/2011 8:21 AM. The bottom right corner shows 'Modified By: Mohan Elango, 1/19/2011 8:21 AM'.

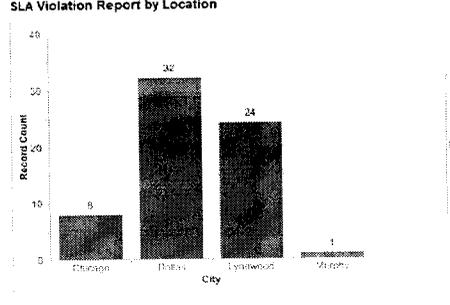
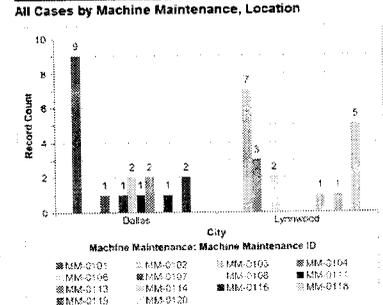
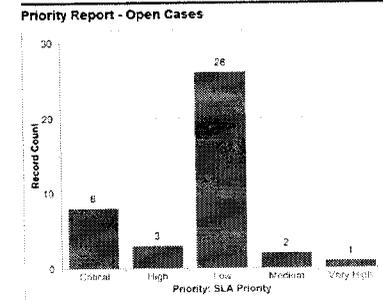
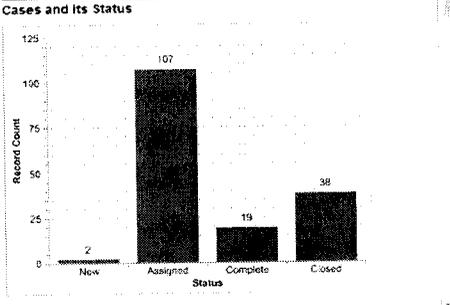
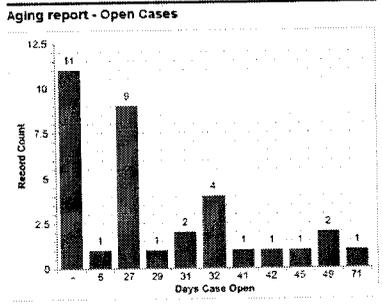
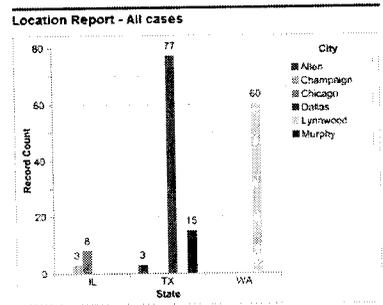
REPORTS:

Dashboard Help for this Page  
**TrackMe**  
 Go to Dashboard List

**View Dashboard**  Edit Clone Refresh

Last refreshed at 3/24/2011 4:22 PM  
 Viewing dashboard as: Mohan Elango

Default Dashboard



### All Case by Machine Maintenance, Location

Report Generation Status: Complete

Report Options:

Summarize information by: City  Show

Time Frame: Date Field  Range  From

Case: Case Number	Location	Machine Maintenance: Address	Machine Maintenance: City	Machine Maintenance: State	Machine Maintenance: Maintenance Days	Machine Maintenance: Description	Machine I
City: Dallas (15 records)							
Machine Maintenance: Machine Maintenance ID: MM-0101 (9 records)							
0-00054	L-0001	Skylark Apartment Complex 1125 E Campbell Rd.	Dallas	TX	20	HVAC Chiller replacement	MM001
0-00002	L-0001	Skylark Apartment Complex 1125 E Campbell Rd.	Dallas	TX	20	HVAC Chiller replacement	MM001
0-00023	L-0001	Skylark Apartment Complex 1125 E Campbell Rd.	Dallas	TX	20	HVAC Chiller replacement	MM001
0-00016	L-0001	Skylark Apartment Complex 1125 E Campbell Rd.	Dallas	TX	20	HVAC Chiller replacement	MM001
0-00113	L-0001	Skylark Apartment Complex 1125 E Campbell Rd.	Dallas	TX	20	HVAC Chiller replacement	MM001
0-00111	L-0001	Skylark Apartment Complex 1125 E Campbell Rd.	Dallas	TX	20	HVAC Chiller replacement	MM001
0-00226	L-0001	Skylark Apartment Complex 1125 E Campbell Rd.	Dallas	TX	20	HVAC Chiller replacement	MM001
0-00003	L-0001	Skylark Apartment Complex 1125 E Campbell Rd.	Dallas	TX	20	HVAC Chiller replacement	MM001

# RACK ME- RECENT ITEMS ENTERED:

SELISOFT Search All Search Mohan Bhangu Help TrackMe

Home Departments TrackMe Search TrackMe Profiles Status Dashboards Reports Screens Request Types SLAs Business Flows Cases TrackMe Users Equipments Parts Local

Recent Items: Finance Complete Account Corrected C-00253 C-00238 C-00236 C-00235 C-00234 C-00233 C-00232

TrackMe Home Link TrackMe Home

Barcodes Home Help for this Page

View: All Got Edit Create New View

### Recent Barcodes

Machine Barcode Name	Barcode	Sub_Id	Location	Address	City	State	Zipcode	Machine Master	Machine Description
MB-0000009	12345678012345	WH-01	L-0001	Skyview Apartment Complex, 1125 E Campbell Rd	Dallas TX	75081	M-0004		GE Water Heater
MB-0000015	4545545	787895	L-0005	1300 Wynstone ST,	Chicago IL	61822	M-0001		HVAC - Carrier
MB-0000008	678998	6786787	L-0004	Foot Hill Commons, 200, 25 th ave west	Dallas TX	75081	M-0004		GE Water Heater
MB-0000010	1234567891012345	HE-N-001	L-0001	Skyview Apartment Complex, 1125 E Campbell Rd	Dallas TX	75081	M-0004		GE Water Heater
MB-0000007	67878989	7878989	L-0004	Foot Hill Commons, 200, 25 th ave west	Dallas TX	75081	M-0004		GE Water Heater
MB-0000034	5432112345	010101010101	L-0005	1300 Wynstone ST,	Chicago IL	61822	M-0005		cooling system
MB-0000022	4657867	7865787	L-0002	Foot Hill Commons, 123 NE 8th Street	Lynnwood WA	98067	M-0005		cooling system
MB-0000021	234234656	678678679	L-0002	Foot Hill Commons, 123 NE 8th Street	Lynnwood WA	98067	M-0005		cooling system
MB-0000020	12345678911111	1234	L-0003	Foot Hill Commons, 123 NE 8th St	Lynnwood WA	98067	M-0005		cooling system
MB-0000003	12567676878	55657567	L-0003	Foot Hill Commons, 123 NE 8th St	Lynnwood WA	99057	M-0005		cooling system
MB-0000025	1122112211221122	Norm	L-0007	833 Morinside Trl	Murphy TX	75094	M-0005		cooling system

# NEW TRACK ME STANDARD PAGE:

SELISOFT Search All Search Mohan Bhangu Help TrackMe

Home Departments TrackMe Search TrackMe Profiles Status Dashboards Reports Screens Request Types SLAs Business Flows Cases TrackMe Users Equipments Parts Local

Recent Items: MB-0000009 Finance Complete Account Corrected C-00253 C-00238 C-00236 C-00235 C-00234 C-00233 C-00232

TrackMe Home Link TrackMe Home

TrackMe Announcement Edit Help for this Page

## New TrackMe Announcement

TrackMe Announcement Edit Save Save as New Cancel

Information \* Required Information

TrackMe Announcement Name:

Type:

Message:

Location Master:

Expiration Date:

---

Case Detail Edit Details Back to Previous Page Set Status To Tenant Keep Me In Loop

Case Number: C-00120 Owner: Madhulika Mittal (Sales User) (C) (Anon)

Request Type: Billing Billing Department:

Description: check my billing Department Assigned: Finance

Priority: Low Priority Modified Date: 9/16/2010 7:43 PM

Machine Manufacturer:  Due Date: 9/16/2010 7:43 PM

Related Order:  Complete Date:

User Assigned: mohan@selisoft.com Issue Date:

---

Location Information

Location: L-0003 Floor Suite:

Address: Foot Hill Commons, 123 NE 8th St

City: Lynnwood

State: WA

Zip/Postal Code: 98067

---

Status Information

Status: New Reset Case Details

---

Tenant Information

# RACK ME- MAINTENANCE HISTORY:

Recent Items

- MH-0014
- Q-00120
- M9-0000009
- Finance
- Complete
- Account Complete
- Q-00253
- Q-00238
- Q-00236
- Q-00235

TrackMe Home Link

MaintenanceHistory  
**MH-0014**

← Back to Edit Custom Object Definitions

[Open Activities \(0\)](#) | [Activity History \(0\)](#) | [Google Docs, Notes, & Attachments \(0\)](#)

Customize Page | Edit Style | Print | View | Help | Feedback Page

### MaintenanceHistory Detail

[Edit](#) | [Delete](#)

Owner: [Behmen Mamber \(Change\)](#)

Maintenance History ID	MH-0014
Location ID	L-0002
Location Description	Lynnwood, WA 98067, Foot Hill Commons, 123 NE 8th Street
Order	O-00047
Machine Master	Dish Washer
Machine Description	GE Dish Washer
Parts ID	Chiller
Parts Description	Chiller coil
Activity	Case Closed
Reason	maintenance complete

## REFERENCES

- [1] Shane Robison, The Next Wave: Everything as a Service, 2007.  
<http://www.hp.com/lhpinfo/execteam/articles/robison/08eaas.html>
- [2] Wei-Tek Tsai, Xin Sun, Janaka Balasooriya, "Service-Oriented Cloud Computing Architecture" in seventh International Conference on Information Technology, 2010, pp 684-689.
- [3] Bhaskar Prasad Rimal, Eunmi Choi, Ian Lumb, "A Taxonomy and Survey of Cloud Computing Systems" in Fifth International Joint Conference on INC, IMS and IDC, 2009, pp 44-51.
- [4] V.Choudary, "Software as a Service: Implications for Investment in Software Development", Proceedings of the 40<sup>th</sup> Hawaii International Conference on System Services.
- [5] Jianfeng Yang, Zhibin Chen, "Cloud Computing Security and issues" in 7<sup>th</sup> International Conference on Computing and security, V2, pg no:89-94
- [6] Thiago Cordeiro, Douglas Damalio, Bob Melandar, "Open Source Cloud Computing Platforms" in the Ninth International Conference on Grid and Cloud Computing, Pg no 366-371.
- [7] Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen and Zhenghu Gong, "The Characteristics of Cloud Computing" in 39<sup>th</sup> International Conference on Parallel Processing Workshops, pg no 275-279.
- [8] Junjie Peng, Xuejun Zhang, Zhou Lei, Bofeng Zhang, "Comparison of several cloud computing platforms", in second International Symposium on Information Science and Engineering, pg no:23-27.
- [9] Xinlei Wnag, Yubo Tan, "Application of Cloud Computing in the Health Information System", in 2010 International Conference on Computer Application and system modeling.V1,Pg no:179-183