Software Requirements Specification

CloudManage System



Manoj Tailor

Sr. Software Engineer

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| May 06,2015 | Version 1.0 | Manoj Tailor | First design document draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Date** | **Comments** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 Overview 1

2. General Description 2

2.1 Product Perspective 2

2.1.1 High Level Design...........................................................................................................................................3

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.2 Hardware Interfaces 3

3.1.3 Software Interfaces 3

3.1.4 Communications Interfaces 3

3.2 Functional Requirements 3

3.2.1 <Functional Requirement or Feature #1> 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.4 Classes / Objects 3

3.4.1 <Class / Object #1> 3

3.5 Non-Functional Requirements 4

3.5.1 Performance 4

3.5.2 Reliability 4

3.5.3 Availability 4

3.5.4 Security 4

3.5.5 Maintainability 4

3.5.6 Portability 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

4. Analysis Models 4

4.1 Data Flow Diagrams (DFD) 4

4.2 State-Transition Diagrams (STD) 4

5. Change Management ProceSS

# 1. Introduction

# 1.1 Purpose

Thepurpose of this document is to present a detailed description of the CloudManage system.It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

# 1.2 Scope

This software system will benamed as **CloudManage**. This software system will provide

Cloud Services Management Platform that fedrates multiple cloud services into one integrated

platform, giving IT centralized management of their public, private and hybrid cloud environments.

By maximizing the user's work efficiency and production the system will meet the user's needs while remaining easy to understand and use.

More specifically, this system is designed to allow an user to Start/Stop/Terminate AWS server instances from easy to use UI with the ability to search and refresh the AWS instances from UI along with overall ingestion. The software will provide fine grained access control over AWS server instances by having user and admin specified roles and permission across differnent cost-center servers. Users will be able to associate parking calender to servers for a specific time period in a day or for days in a week. And this feature will provide the cost management and cost reduction as per cost-center. Parking Calender will be created by Admin roles and then users will be associate the parking calender to server instances using software UI.

# 

# 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| EC2 | Elastic Compute Cloud |
| AWS | Amazon Web Service |
| RDS | Relational Database Service |
| Parking Calender | Echeduling time to start/stop the amazon ec2 instances |
| AD | Active Directory |
| Cost-center | This will represent the project name or a name assigned to a project for their Aws costing. |
| Reviewer | A person that examines an article and has the ability to recommend approval of the article for publication or to request that changes be made in the article. |
| SDK | Software Development Kit |

# 1.4 Overview

The next chapter, the Overall Description section, of this document gives an overview of the

functionality of the product. It describes the informal requirements and is used to establish a context

for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the

developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for

different audiences and thus use different language.

# 2. Overall Description

# 2.1 Product Prespective

CloudManage is web based system implementing client-server n-tier model, aiming towards

users who want to create AWS EC2 servers for their respective projects and after creation

they want to manage the created servers like to start/stop/terminate the servers and then

creating new servers as per their requirement.

This software will be focused towards to provide easy to use UI to manage the AWS EC2

server instances as per their cost-center by having a robust security mechanism.

CloudManage should be user-friendly, 'quick to learn' and reliable software for the above

purpose.

CloudManage is intended to be a stand-alone product and in exception it would only depend

on the Amazon AWS SDK. It should run on both UNIX and Windows based platform.

The Followings are the main features that are included in CloudManage:-

* **Cross platform support**: Offer operating support for most of the known and commercial

operating systems.

* **Global Role :** The system will have a default admin global role and then will

allow this admin to create different global roles as per requirement by giving

permissions for different cost-center and other permissions.

* **Cost-center Role :** The System will allow the admin role user to create the different cost-

center roles as per requirement by giving permissions for different operations to

perform. And then if a user have specific cost-center role associated, only be able to

access that cost-center servers.

* **Search:** Search will be a local search engine based on instane name, id , private ip

address etc.

* **Managing Servers:** Authenticated user would be able to manage the server like

staring/stopping/terminating the server based on assigned permissions to their roles and

would be able to view the details of a perticular server along with tagging the server

instances.

* **Number of users being supported by the system:** Though the number is precisely not

mentioned but systmem will be able to support all AD users.

* **Parking Calender :**  The autenticated user would be able to create the parking calender

and to associate them with particular AWS EC2 server instances.

# 2.1.1 High Level Design :

The following diagram shows the High level design for CloudManage application.

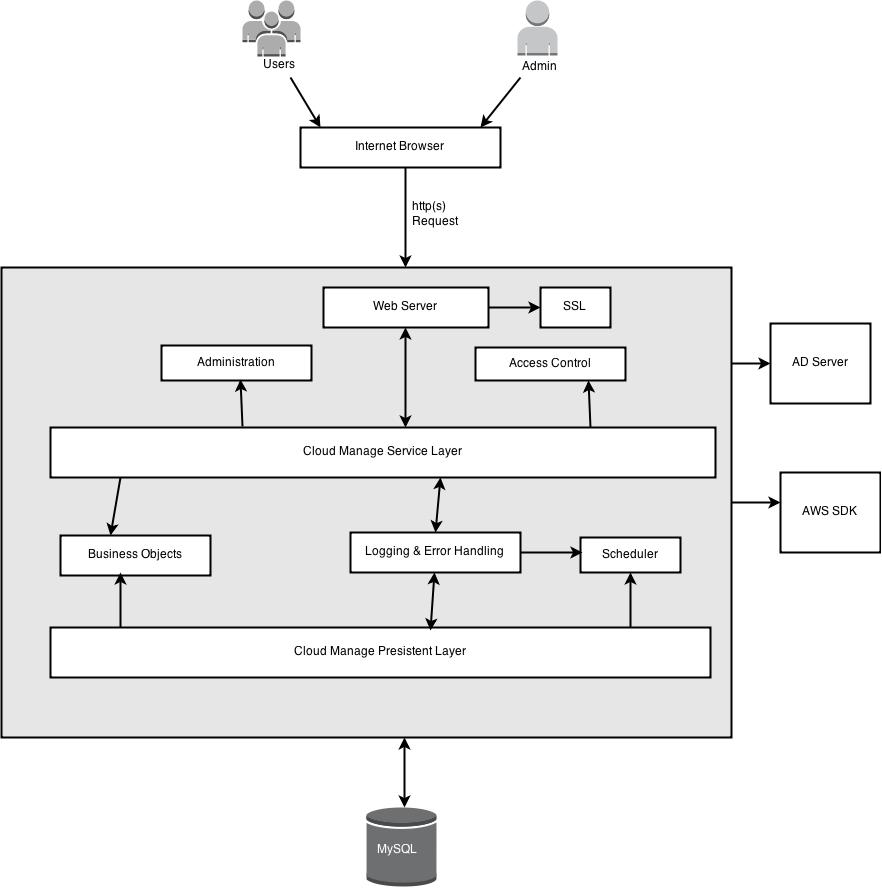


Fig 1- High Level Design of CloudManage

The system design utilizes web based multi-tier system architecture. The admin and other

users will access the internet broswer to send the http(s) request to web server by using SSL

(Secure Socket Layer) and then web server will forward the request to the deployed

CloudManage Application.

CloudManage Service layer will use access control mechanism to aunthenticate and

authorise the users. Business objects will be used to with both service layer and perisistent

layer. Logging and Error handling will be responcible to creating logs of every activity in

both file and database and to view the audit reports. The whole application would be depend

on the AWS SDK to perform actions on AWS EC2 servers.

Scheduler would be used to handle the parking calender and would take care of that by using

the dababase entries. All the required data to perform actions and parking calender activity

the Relatonal Database will be used.

# 2.2 Product Functions

With this software system an authenticated user will only be able to access and view details

of servers for which authorization provided by the admin user. If suppose user 'X' is

authorized to cost-center 'Y' then user 'X' should be able to view and access all server of 'Y'

cost-center. User 'X' should also be able to associate parking calender with cost-center 'Y'

servers.

CloudManage should support the following use-cases :

|  |  |  |
| --- | --- | --- |
| **Classes of Use Cases** | **Use Cases** | **Description of Use Cases** |
| Use Case Related to Authentication | Login | Login into CloudManage using AD authentication |
| Use Case Related to Managing Global Role | Manage Global Role | Creating new global role by using dafault admin role |
| Use Case Related to Managing Cost-center Role | Manage Cost-center Role | Creating new cost-center roles by using default admin role |
| Use Case Related to Assgining Global Role | Assign Global Role | Assigning users to global roles |
| Use Case Related to Assigning Cost-center Role | Assign Cost-center Role | Assigning users to cost- center roles |
| Use Cases Related to Creating Parking Calender | Create Parking Calender | Creating Parking Calender to associate them with server instances |
| Use Cases Related to Associating Parking Calender | Associate Parking Calender | Any user can associate the parking calender to their respective cost-center server instances |
| Use Cases Regarding to Executing Parking Calender | Executing Parking Calender | Schedular will execute the associated parking calender to start and stop the respective servers |
| Use Cases Regarding Audit Reports | Audit Reports | Fetching details from DB and showing audit reports per server-instance to users |

# 2.3 User Characteristics

There are two types of users that interact with the system: cost-center user and admin user.

Each of these two types of users has different use of the system so each of them has their

own requirements.

The Cost-center users can be sub categorized to cost-center user and multiple cost-center

user. The cost-center user will only be able to perform actions to which he is authorized to

perform on his associated cost-center while mutiple cost-center user will also be able to

perform the more actions compared to cost-center user across multiple cost-center to which

he is associated.

The admin user will be super user and he will have all the rights to perform any actions like

managing global and cost-center roles and assigning global and cost-center roles to user,

creating parking calender, deleting parking calender, viewing audit reports.

# 2.4 Assumptions and Dependencies

One assumption is that AD Server must be working as it is dependent on AD Server to

authenticate any user and system will not be usable if AD server is not working or there is no

connectivity in between AD Server and Web Server.

Any user should have internet connection to use this software application.

This application is dependent on AWS SDK to perform actions on AWS EC2 server instances.

# 3. Specific Requirements

# 3.1 External Interface Requirement

# 3.1.1 User Interfaces

A first-time user of the mobile application should see the log-in page when he/she opens the

application. After login into application user will see the dashboard in which they can see the

details of server instances and start/stop/terminate the servers and can search a server by

instance id , private ip , dns name etc.

If an admin user is login than manage roles and assign roles view can be accessed. In manage

role view he/she will be able to manage the global and cost-center roles and in assign roles

view he/she will be able to assign the global and cost-center roles. There would also be a

parking calender view by which he/she will be able to create new parking calenders. For the

users view of parking calender, he/she will be able to associate created parking calender to

any server instances.

Any user will also have the tag view by which he/she can associate tags to particular servrer

instances and then can manage the tags like adding more tags, updating previous tags value,

deleting tags. In the audit server view the admin user will be able to the audit report on a

particular server to diagnose what actions have been performed on this server and by whom

those actions are performed.

# 3.1.2 Hardware Interfaces

CloudManage dont't have any designated hardware, it does not have any direst hardware

interfaces. Hardware connection from Web Server to AD and database server is managed by

the underlying operating system on the Web Server.

# 3.1.3 Software Interfaces

The CloudManage application communicates with the AD server in order to authenticate an

user, with database in order to store audit and parking calender details and with Amazon EC2

server using Amazon java Sdk to access , manage and view details of EC2 servers. The

communication between the CloudManage and database cosists of operation concerning both

reading and modifying the data.

# 3.1.4 Communication Interfaces

The communication between the different parts of the system is important since they depend

on each other. However, in what way the communication is achieved is not important for the

system and is therefore handled by the underlying operating systems for CloudManage.

# 3.2 Functional Requirement

This section includes the requirements that specify all the fundamental actions of

CloudManage.

**Use Case 1**: Login

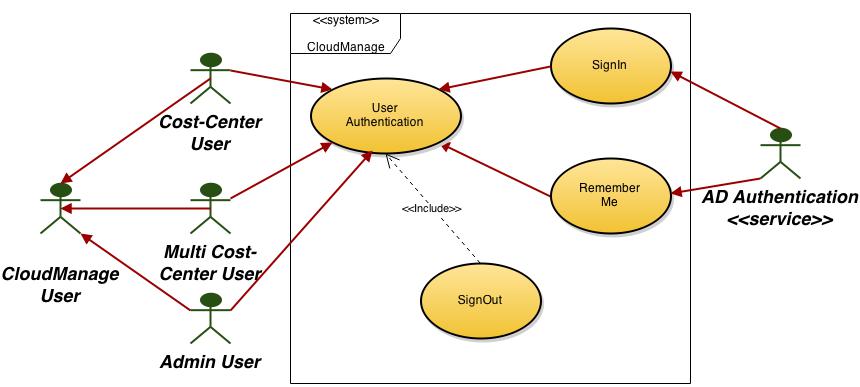


Fig 2: Use Case For User Authentication

|  |  |
| --- | --- |
| **Item** | **Description** |
| Context of Use | Only authenticated users are allowed to gain access to the CloudManage resources. The CloudManage relies on the user ID maintained in the corporate AD. |
| Scope | The CloudManage users identify themselves only once for a complete online session. In the case that an idle time-out occurs, the users must reidentify themselves. The identified CloudManage users are able to access only the server instance and information that they are authorized for. |
| Primary User | CloudManage Cost-center, Multi Cost-center and admin user. |
| Preconditions | The CloudManage user is an authorized user. Standard corporate AD user ID must be provided to each employee. Every employee has access to the corporate intranet. No guest account. |
| Description | 1. User enters the portal URL. 2. If the customization parameter [remember login] is set, then automatically login the user and provide a session ID. 3. If first time user, prompt for LDAP user ID and password. 4. User enters previously assigned user ID and password. 5. Information is passed to Access Manager for validation. 6. If authentication passes, assign session ID and continue. 7. If authentication fails, display error message, return user to login page. |