PSA IT Division

*PSA Reminder 365*

*Technical Requirement and Design Specification*

*0.1*

*{Confidential}*

|  |  |
| --- | --- |
| Title | PSA Reminder 365 |
| Subject | Technical Requirement and Design Specification |
| Version | 0.1 |

© 2009 PSA Corporation Limited. All rights reserved.

1. **Reference to Master Document**

|  |  |
| --- | --- |
| File server |  |
| URL |  |

1. **Reviewers and approvers**

|  |  |  |
| --- | --- | --- |
| Reviewed by: | | |
| Full name/Designation/Company | Signature | Date |
| <Reviewer>  <designation> |  |  |

|  |  |  |
| --- | --- | --- |
| Approved by: | | |
| Full name/Designation/Company | Signature | Date |
| <Approver>  <designation> |  |  |

1. **Version History**

|  |  |  |
| --- | --- | --- |
| Version | By/When | Summary |
| 0.1 | Venkatesh N | First release. |
|  |  |  |

♦ *End of Document Control Record* ♦

**Contents**

[1 Introduction 5](#_Toc502823814)

[1.1 Definitions, Acronyms and Abbreviations 5](#_Toc502823815)

[1.2 References 5](#_Toc502823816)

[2 Requirement Study 6](#_Toc502823817)

[2.1 Service Level Requirement 6](#_Toc502823818)

[2.1.1 Availability and Reliability 6](#_Toc502823819)

[2.1.2 Disaster Recovery 6](#_Toc502823820)

[2.1.3 Degradation Mode 6](#_Toc502823821)

[2.2 Non-Functional Requirement 6](#_Toc502823822)

[2.2.1 Performance/Response Time 6](#_Toc502823823)

[2.2.2 Throughput/System Load 6](#_Toc502823824)

[2.3 User Access 6](#_Toc502823825)

[2.4 Application Security 6](#_Toc502823826)

[3 Design Goals 7](#_Toc502823827)

[3.1 Primary Design Goals 7](#_Toc502823828)

[3.2 High Level Solution Overview 7](#_Toc502823829)

[3.3 Alternative Solution Approaches 8](#_Toc502823838)

[3.4 Selected Solution 8](#_Toc502823839)

[3.5 Assumptions 8](#_Toc502823840)

[3.6 Implementation Strategy 8](#_Toc502823841)

[3.7 Solution Risks 8](#_Toc502823842)

[4 Data Architecture 8](#_Toc502823843)

[4.1 Logical Data Model 8](#_Toc502823844)

[4.2 Data Replication 8](#_Toc502823845)

[5 Application Architecture 9](#_Toc502823846)

[**Angular Application – Client-Side Component** 9](#_Toc502823847)

[5.1 System Overview 9](#_Toc502823848)

[5.2 System Interface and Linkages 10](#_Toc502823849)

[5.3 Data Flow Diagram 11](#_Toc502823850)

[5.4 Design Solutions 11](#_Toc502823851)

[5.4.1 Authentication 11](#_Toc502823852)

[6 Dependencies 12](#_Toc502823853)

[6.1 List of Infrastructure Dependencies 12](#_Toc502823854)

[6.2 List of External System Dependencies 12](#_Toc502823855)

[6.3 List of External Systems Depending on this System 12](#_Toc502823856)

[7 Technical Architecture 12](#_Toc502823857)

[7.1 Deployment View 12](#_Toc502823858)

[7.2 Hardware/Software Planning and Sizing 13](#_Toc502823859)

[7.2.1 Applications Server 13](#_Toc502823860)

[7.2.1.1 Server 13](#_Toc502823861)

[7.2.1.2 Memory 13](#_Toc502823862)

[7.2.1.3 Disk Space 14](#_Toc502823863)

[7.2.1.4 Required Software Packages 14](#_Toc502823864)

[7.2.1.5 Middleware and Usage Requirement 14](#_Toc502823865)

[7.2.2 PCs 14](#_Toc502823866)

[7.2.2.1 Specifications 14](#_Toc502823867)

[7.2.2.2 Required Software Packages 14](#_Toc502823868)

[7.2.3 Archival 15](#_Toc502823869)

[8 Deviations, Violations and Exceptions 15](#_Toc502823870)

# Introduction

## Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| **No.** | **Definitions, Acronyms and Abbreviations** | **Description** |
| 1 | API | Application Programming INterface |
| 2 | DR | Disaster Recovery |
| 3 | OWASP | Open Web Application Security Project |
| 4 | SPA | Single Page Application |
| 5 | AJAX | Asynchronous Javascript and XML |
| 6 | UI/UX | User Interface/User Experience |

## References

|  |  |  |
| --- | --- | --- |
| **No.** | **Reference** | **Description** |
| 1 | User requirement Specifications |  |
| 2 | Use Cases |  |

# Requirement Study

## Service Level Requirement

### Availability and Reliability

The system should have minimal downtime. The system should be highly reliable. There should be no loss or corruption of data.

### Disaster Recovery

The solution will be implemented on a DR site. This site will be inactive and will have to be made active during failover process.

### Degradation Mode

Not Applicable

## Non-Functional Requirement

### Performance/Response Time

|  |  |  |
| --- | --- | --- |
| **No.** | **Item** | **Required performance/response time** |
| 1 | Enquiry | Shall not exceed three (3) seconds |
| 2 | Create/Update/Delete | Shall not exceed three (3) seconds |
| 3 | Report/Dashboard | Shall not exceed three (3) seconds |

### Throughput/System Load

|  |  |  |
| --- | --- | --- |
| **No.** | **Item** | **Required throughput/Expected system load** |
| 1 | Database Throughput | 10 |
| 2 | *Number of concurrent users* | 20 |
| 3 | *Number of concurrent calls* | 20 |

## User Access

|  |  |  |
| --- | --- | --- |
| **No.** | **User** | **Access Requirement** |
| 1 | HR/Admin | Ability to manage users and groups |
| 2 | Group users | Ability to set and view reminders within their groups |

## Application Security

The setup should be adequately protected against the following types of threats to ensure authenticity, confidentiality, integrity, availability, non-repudiation and accountability of data, service and access:

* Malware
* Denial-of-service
* Spoofing
* Replay, Interception and modification of data
* Insider attacks
* Password guessing attacks
* Web vulnerability attacks including OWASP top ten

Weak encryption attacks

The setup should adhere minimally to PSA IT Security Standards, with emphasis on following sections:

* Communications and Operations Management
* Access Control
* Information Systems Acquisition, Development and Maintenance
* Compliance

The setup should adhere minimally to NIST SP800-82 Guide to ICS System Security.

The System shall include audit trail and audit log capabilities for Systems and database operational activities. The audit trails shall not affect the Systems performance significantly where it impact users’ experience.

Timeout and logout features shall be set for non-active session for web and client server access. The timeout period feature shall be configurable. As far as possible, the System shall warn the users of impending timeouts.

# Design Goals

## Primary Design Goals

|  |  |
| --- | --- |
| **No.** | **Primary Design Goals** |
| 1 | *The role based user module should be highly configurable such that new roles can be created and the privileges can be configured for those roles* |
| 2 | *The reminder module should allow authorized users to create, search via keyword(s), filter, view, modify and delete the reminder records of their own group.* |
| 3 | *All changes to different modules should be auditable* |
| 4 | *Multiple users can upload files to the system at the same time without slowing down the system* |
| 5 | *User should be able to see views for different reminder modules in the dashboards* |
| 6 | *Maker and Checker function for contracts reminder should be configurable* |
| 7 | *Only users with proper authorization should be able to carry out the specified actions* |
| 8 | *Reminder notifications should be triggered on a timely basis and recorded* |
| 9 | *A centralized identity provider should exist* |

## High Level Solution Overview

The new Reminder 365 application that allows users to manage and monitor the expiry dates of asset, certificates and contracts. When nearer to date, application should be smart enough to consolidate the expiring items of the same category and send an email to notify the respective stakeholders of that user group.

The proposed solution will be implemented using a backend application and frontend single page application. All the UI will be implemented as a thin client and business rules and process on the server side. UI will send a request for data to server side using AJAX calls and the response will be in JSON data format.

The system shall be deployed as a separate web application in the current system. The system shall also make use of the current notifications system for notifications. There shall be proper auditing for all the operations the same way it’s done in the other modules. There shall be a proper implementation to help retrieve the information and backtrack the reason in the case of any issue occurs during system usage.

Following modules will be implemented –

* Authentication and Authorization Module
* Contract Reminder Module
* Asset Reminder Module
* Staff Reminder Module
* Notifications
* Dashboard Module
* Settings Module

## Alternative Solution Approaches

An alternative way to build the solution would be to build multi page application with the web pages being built as jsp pages rendered on the server side.

## Selected Solution

We are going with the approach of building APIs and SPA in the frontend as this approach gives following advantages -

1. Applications are extremely responsive as only portions of a page get updated when action are done on the page
2. Possibility of caching data on the browser
3. Easier to make mobile application in future as the same backend API can be used
4. Much of the data validation can be done in frontend
5. Frontend development becomes more streamlined and efficient

## Assumptions

|  |  |  |
| --- | --- | --- |
| **No.** | **Assumptions** | **Remarks** |
| 1 | Users will use modern web browsers with javascript enabled |  |
|  |  |  |

## Implementation Strategy

## Solution Risks

Following are the risks of using this solution

1. Javascript should be enabled for all clients
2. Some native features like cancel action on the browser or back button might not behave as they do in multi page applications
3. Initial load of page is slow

# Data Architecture

## Logical Data Model

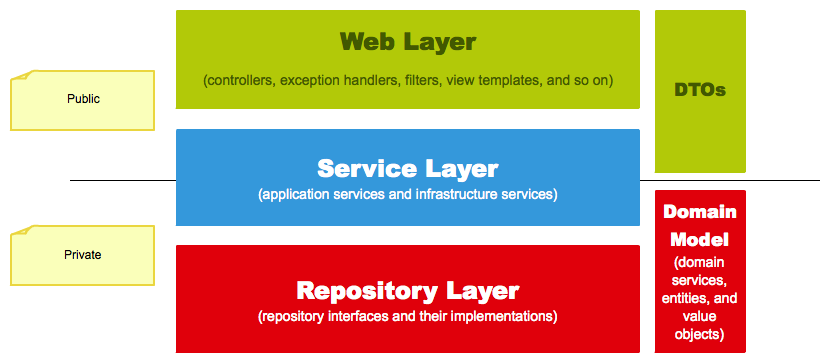
Refer to the ER diagram

## Data Replication

Data will be replicated between Prod and DR using master slave mode.

# Application Architecture

The Proposed new backend systems will 3-Tier web application architecture as shown below



**Web Layer:**

Web layer will be implemented using SPRING framework. It will consist of controller classes, exception handlers

1. Controller classes – Responsible for controlling the flow of the application
2. Exception handlers – responsible for handling exceptions

**Service Layer**

1. Responsible for integration with external systems
2. Responsible for business rules and transaction processing

**Repository Layer**

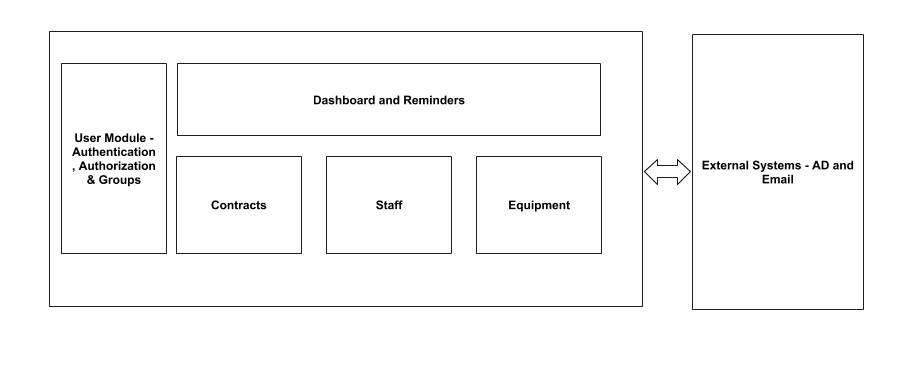
DAO – Data access objects for database persistence and querying.

### **Angular Application – Client-Side Component**

The client will be implemented as Web thin client using Angular 4 JavaScript framework. Client side program will be used for rendering the display using HTML5, CSS3, and Angular4 using data from Server side in JSON format. Client side code will make API call to the server using JSON

## System Overview

Following image depicts the overall system and its modules

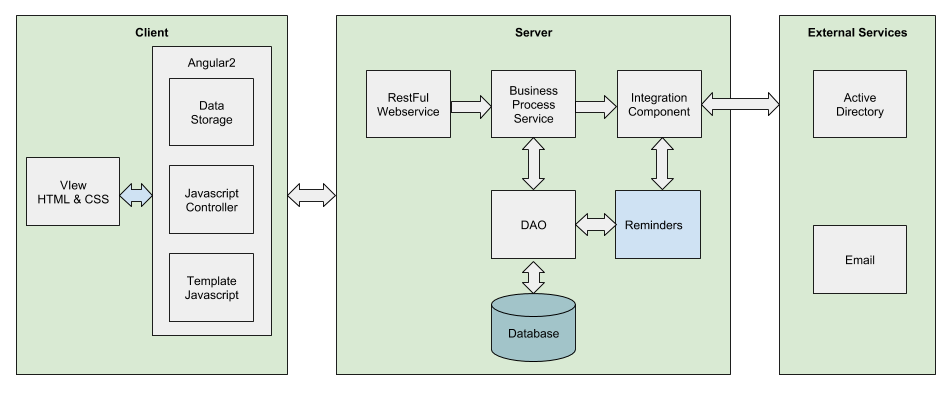


|  |  |  |
| --- | --- | --- |
| **No.** | **Modules** | **Description** |
| *1* | *Contracts* | Allow authorized users to create, search via keyword(s), filter, view, modify and delete the contract reminder records of their own group. |
| *2* | *Staff* | Allow authorized users to create, search via keyword(s), filter, view, modify and delete the staff reminder records of their own group. |
| *3* | *Asset* | Allows authorized users to create, search via keyword(s), filter, view, modify and delete the asset reminder records of their own group. |
| *4* | *Users and Groups* | Login and Logout, Manage User Group and Role,Manage User within User Group |
| *5* | *Dashboard and Reminders* | Allow user to have an overall view of the status of the items that he is been assigned to monitor (i.e member of the user group for that particular module). This feature shall base on a schedule to periodically perform a daily check on the expiring items. Those information of the expired and expiring items shall be generated into a spreadsheet and send to the users under the specified user group via email. |

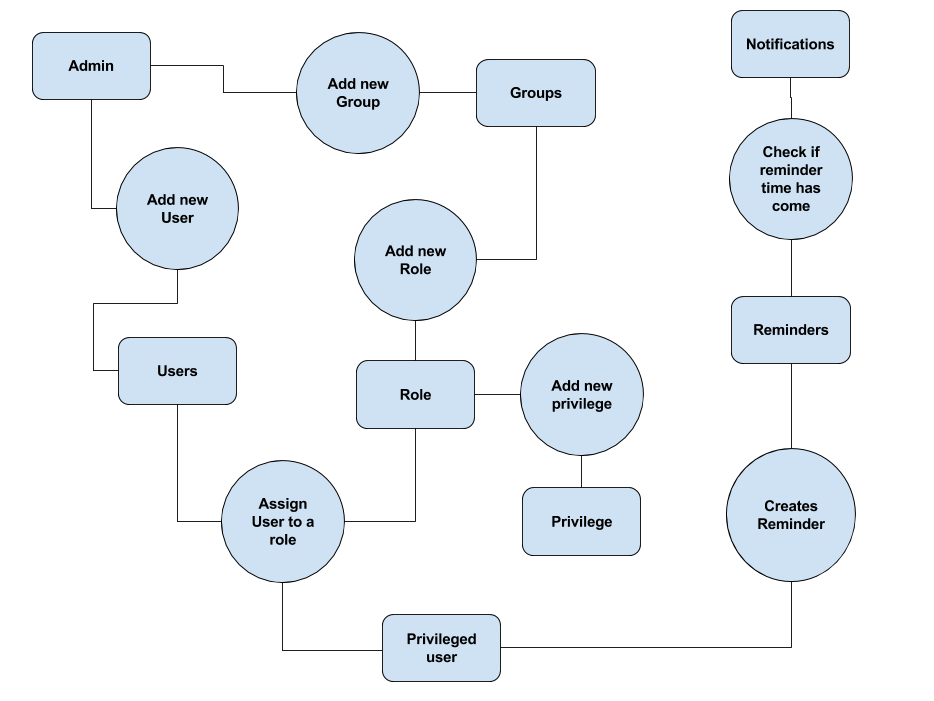
## System Interface and Linkages

The system will be talking to Active Directory to authenticate the user using login and password. No other external system will be integrated.

Following image depicts the interaction between various layers -



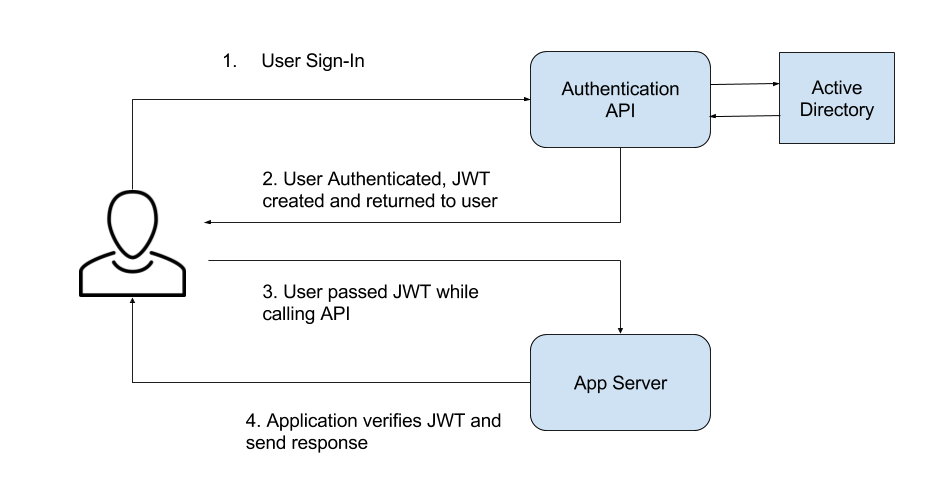
## Data Flow Diagram



## Design Solutions

### Authentication

Following image depicts how authentication will work -



# Dependencies

## List of Infrastructure Dependencies

|  |  |  |
| --- | --- | --- |
| **No.** | **Infrastructure** | **Impact to System** |
| 1 | Underlying Virtualization provider | The software will become inaccessible |
| 2 | Network | * The software will become inaccessible * Might lead connectivity issues between Production and DR leading to loss of sync of data |
| 3 | DNS/Load balancers | It might lead to request not reaching the right web servers leading to downtime |

## List of External System Dependencies

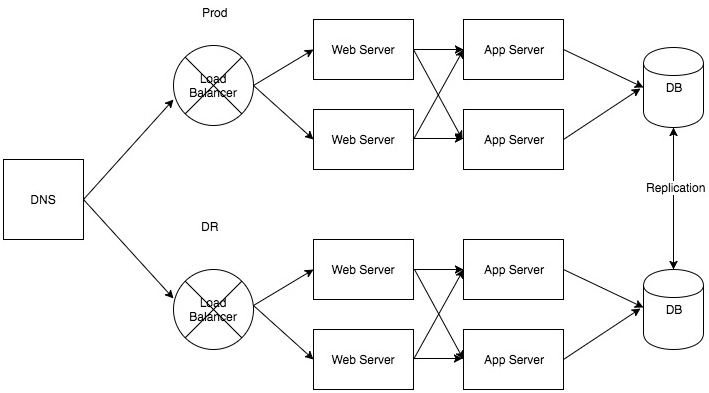
|  |  |  |
| --- | --- | --- |
| **No.** | **External System** | **Impact to System** |
| 1 | Active Directory | Authentication will fail |
| 2 | External Backup | Backups will fail |

## List of External Systems Depending on this System

No external system is depending on this system.

# Technical Architecture

## Deployment View



There will be two sites - Production and DR. On each site, there will be web servers which will route the traffic to the App servers. There will be a database at each site. Data will be replicated between the sites. At any point, one will be the master and the other the slave.

The load can be balanced between multiple webservers by creating multiple a-records with the same name as well. However, it would not be able to detect any failures of downstream servers.

During failover, the master-slave roles have to be switched. The web URL has to be pointed to the right site. Failovers will not be automatic.

Note - The diagram is for illustration of deployment view. For actual number of servers, please refer to Hardware Planning.

## Hardware/Software Planning and Sizing

### Applications Server

### Server

|  |  |  |
| --- | --- | --- |
| **No.** | **Purpose** | **Specification** |
| *1* | *2 servers to run backend application and 1 server for web server* | *Operating System – Linux preferably Debian flavour*  *Processor - >= 2.4 GHZ* |

### Memory

|  |  |  |
| --- | --- | --- |
| **No.** | **Server** | **Memory Needed** |
|  | Example: |  |
| *1* | *2 servers to run WLS instances* |  |
|  | *- WLS instance* | *Each WLS instance - 4 GB heap* |
|  | *Total* |  |
| *2* | *1 server to house webserver* |  |
|  | *- nginx instance* | *Each nginx – 2GB heap* |
|  | *Total* |  |

### Disk Space

|  |  |  |
| --- | --- | --- |
| **No.** | **Item** | **Estimated Disk Space** |
| *1* | *2 servers to run WLS instances* |  |
|  | *- Application Log* | *Total disk space required per day = 50 MB*  *Number of day to retain online = 30*  *Total disk space to retain offline = 1.5 GB* |
|  | *- Application Data File* | *Total disk space required per day = 50 GB (5000 files can be stored considering each file is of size 10MB)* |
|  | *- Database server* | 10GB of disk space |
| 2 | 1 server to run *webserver* |  |
|  | *- Application Log* | *Total disk space required per day = 50 MB*  *Number of day to retain online = 30*  *Total disk space to retain offline = 1.5GB* |
|  | *- Application Data File* | *Total disk space required 5 GB* |

### Required Software Packages

|  |  |  |
| --- | --- | --- |
| **No.** | **Software Package** | **Product and Version** |
| *1* | *Java JDK* | *Java JDK 1.7 and above* |
| 2 | MySQL server | MySQL 5.6 and above |
| 3 | WebLogic server | WebLogic 12c |
| 4 | nginx | nginx 2 and above ( for angular application) |

### Middleware and Usage Requirement

|  |  |  |
| --- | --- | --- |
| **No.** | **Middleware** | **Descriptions** |
| *1* | *Database* | *MySQL*  *Estimated number of table = 20*  *Estimated number of record size = 2kb*  *Estimated access frequency = Not Defined*  *Estimated retention period = Not Defined*  *Backup and archival requirement* |
| *2* | *Web Server* | *As per master TRDS* |
| *3* | *LDAP* | *As per master TRDS* |

### PCs

### Specifications

|  |  |  |
| --- | --- | --- |
| **No.** | **Purpose** | **Specifications** |
|
| *1* | *20 x high end PC for users to run browser based web application* | *Operating System – Windows 7 or above*  *Processor - >= 2.4 GHZ*  *Memory – 4 GB*  *Hard Disk Space – 40 GB*  *Monitor Display – 19” Dual Monitor* |
|  |  |  |

### Required Software Packages

|  |  |  |
| --- | --- | --- |
| **No.** | **Software Package** | **Product and Version** |
| *1* | *Browser - IE/Chrome* | **IE Version** – 11 and above  **Chrome Version** – 50 and above |

### Archival

Not applicable

# Deviations, Violations and Exceptions

No deviations, violations and exceptions identified yet.

♦ *End of Document* ♦