

Software Requirements Specification

SEMiOne – PROSemi

Document Information

Reference	BRD	Version No.	1.0
Release Date	18-01-2021	Total No. of pages	25
	Name	Designation	Date
Prepared by	Jothikrishnan	Software Developer	11-01-2021
Reviewed By	Krishna	Business Analyst	15-01-2021
Approved by	Ram	Project Manager	18-01-2021

Document Amendment Record

S. No.	Rev. From	Rev. To	Date of Change	Changes Made
1	1.0	1.0	18-01-2021	Initial Draft

Document Organization

#	Chapter	Contents
1.	Introduction	Provides a brief description of the document
2.	System Architecture	Proposed system architecture for PROSemi
3.	System / Application Overview	Provides detailed description of the project scope, features, and notations
4.	Functional requirements	Details the important functional requirements as numbered requirements in each module
5.	Non-Functional/ Other Requirements	Security, Reliability constraints, etc.
6.	Acceptance criteria	Criteria required by the client to accept the project
7.	References	Bibliography of documents referred during the preparation of this document

Table of Contents

1. Introduction.....	6
1.1. Purpose.....	6
1.2. Scope	6
1.3. Intended Audience.....	6
1.4. Acronyms & Abbreviations	6
1.5. Document Organization	Error! Bookmark not defined.
2. System Architecture	7
3. System / Application Overview	12
3.1. Overview	12
3.2. Features.....	12
3.3. Assumptions & Dependencies.....	13
4. Functional Requirements.....	14
4.1. End user site	14
4.1.1 Login	14
4.1.2 General Requirements	14
4.1.3 My Profile	15
4.1.4 End User Logout	16
4.2. Admin site	16
4.2.1 Login	16
4.2.2 Master Menu	17
4.2.3 My Profile	17
4.2.4 Master Data Entry	18
4.2.5 Release Pending Tickets	19
4.3. Interface Requirements	Error! Bookmark not defined.
4.3.1 Software Interfaces	19

4.3.2	User Interfaces	20
4.3.3	Communication Interfaces	22
5.	Non-functional / Other Requirements.....	23
5.1.	Security Requirements.....	23
6.	Acceptance Criteria.....	23
7.	References.....	23

Table of Figures

Figure 1: System Architecture 7

Figure 2: Application architecture for PROSemi 10

Figure 3: Reserve Use of Resources..... 21

Figure 4: Reports 21

Figure 5: Search Tickets 22

1. Introduction

PROSemi is an intranet application that performs reservation of resources required for testing. **SEMiOne** reserves the resources using this intranet application. This application would be used by various users of the **SEMiOne**.

1.1. Purpose

The purpose of this document is to list the system requirements in detail for **PROSemi** and modularize them.

1.2. Scope

The scope of this document is to list all the functionalities/features envisaged for the new automated **PROSemi** system. **PROSemi** should be modelled by referring to the BRD provided by the **SEMiOne**.

1.3. Intended Audience

Intended Audience	Tick whenever Applicable
Project Manager	✓
Project Leader	✓
Project Team Member	✓
Client	✓
PMG	✓
Configuration Controller	✓
Test Engineer	✓

1.4. Acronyms & Abbreviations

Acronyms	Description
BRD	Business Requirement Document
DAL	Data Access Layer
LDAP	Lightweight Directory Access Protocol
PMG	Process Management Group
PO	Purchase Order
SMTP	Simple Mail Transfer Protocol

2. System Architecture

The following diagram describes the proposed system architecture for the **PROSemi**.

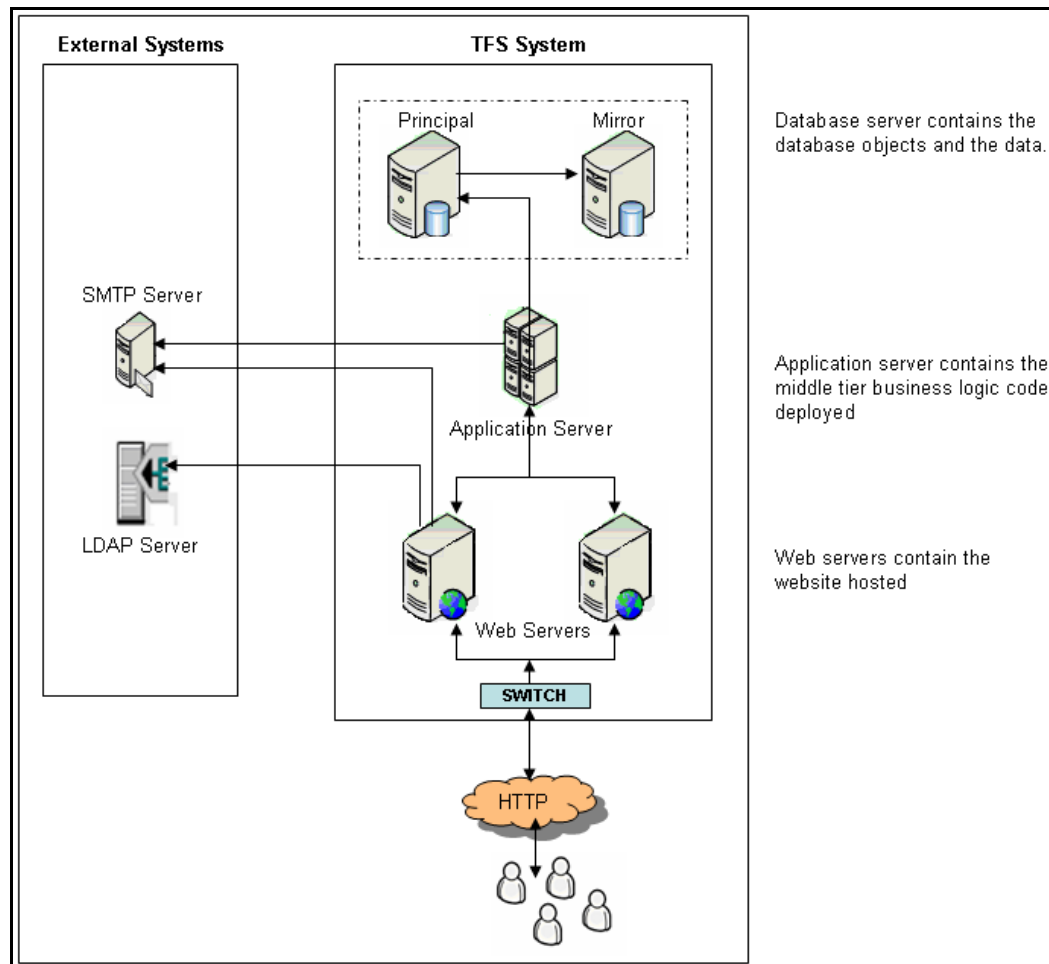


Figure 1: System Architecture

2.1. PROSemi System

2.1.1 Web Server

- **Hardware Requirements**

HP ProLiant DL360 Generation 5 (G5) **OR** BL460c

- **Software Requirements**

Software	C# and ASP.NET
Web Server	Internet Information Server 5.1 & 6.0
Environment	.NET Framework 2.0
IDE	Visual Studio 2005
Targeted Browser	IE 6.0 and above

- **Applications executed on this server**

Entire code related to the **PROSemi** Intranet site should be deployed on this server. This server has all the web pages related to the presentation layer.

- **Presentation Layer**

The users of **PROSemi** should access the application through the presentation layer. The user interfaces of the application are web pages designed in C# and ASP.Net.

2.1.2 Database Server

All the Database objects should be created in this server and the application interacts with this server for data retrieving and storage.

- **Hardware Requirements:**

HP ProLiant DL500 Servers **OR** BL460c

- **Software Requirements:**

- SQL Server 2005 Enterprise Edition
 - SQL SERVER 2005 Reporting Services (SSRS)

- **Other Implementations:**

SQL Server implementation in this case should be **SQL Server Mirroring**. This is to provide high availability.

- **Alternative:**

The proposed alternative for SQL Server Mirroring is **Log Shipping**. The Log Shipping also provides high availability but unlike SQL Server Mirroring, it does not come with automatic failover.

- **Applications executed on this server:**

This server has the **PROSemi** database and the stored procedures. All backend jobs (if any) should be executed on this server.

2.1.3 Application Server

- **Business Logic Layer**
- **Data Access Layer**

The Data access layer should be implemented using the ADO.Net 2.0 features.

2.2. External Systems

2.2.1 SMTP Server

This server should be used for email notifications from and to for the **PROSemi** application.

2.2.2 LDAP Server

This server contains the LDAP data right from where the user can be authenticated and can share data across the systems wherever applicable.

The following diagram describes the proposed application architecture for the **PROSemi**.

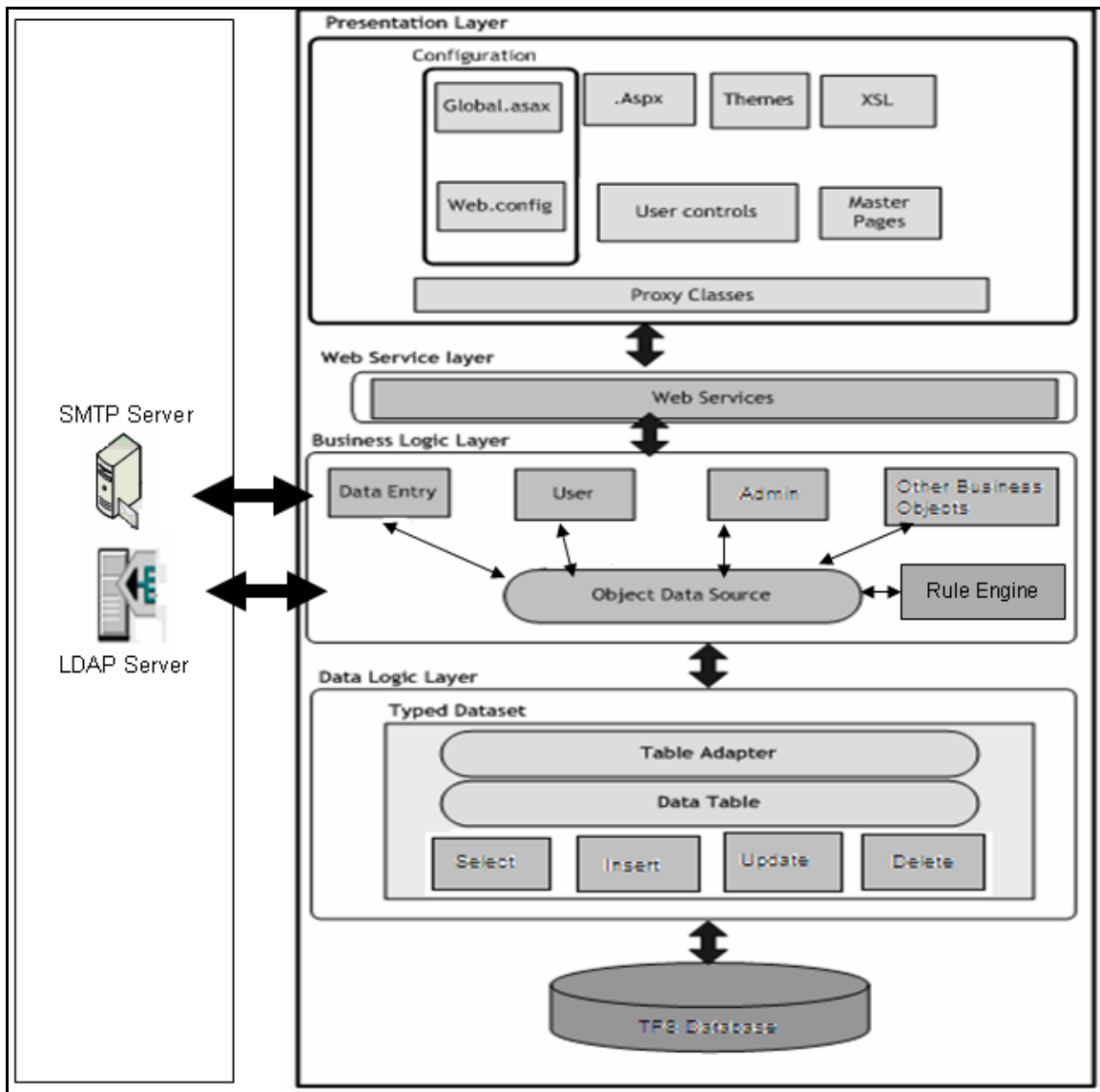


Figure 2: Application architecture for PROSemi

2.3. Presentation Layer

The users of **PROSemi** should access the application through the presentation layer. The user interfaces of the application are web pages designed in C# and ASP.NET.

2.4. Business Logic Layer

The Business Logic Layer should form the core layer of the application where the business rules and data validations should be implemented. The business layer should receive the events/requests from the user interface and process them by the object data source from the DAL. The interface for the web services layer should be via wrapper classes.

2.5. Data Access Layer

The Data Access Layer (DAL) should not be using ADO.NET implementations. Actual implementation should be decided at a later point in time and tools like Enterprise Library Data Access Application Block can be considered. This application block should be used in a variety of situations, such as reading data for display, passing data through application layers, and submitting changed data back to the database system. The application block includes support for both stored procedures and in-line SQL. Common housekeeping tasks, such as managing connections and creating and caching parameters, are encapsulated in the application block's methods. In other words, the Data Access Application Block provides access to the most often used features of ADO.NET in simple-to-use classes.

Certain key database operations like transaction management, concurrency issues, connection pooling, multi-databases are in-built features of **Enterprise Library** based DAL.

3. System / Application Overview

PROSemi is primarily a web-based application, reserving the resources for semiconductor testing.

Following are the main objectives:

- Proposed **PROSemi** system should eliminate the limitations of the existing system.
- Must be able to extend for future enhancements with any other components or new features.
- Should have role-based user access.
- Must generate reports from one central repository.
- Should have an audit track feature to track the changes made for the data.

3.1. Overview and Review

PROSemi is envisaged to be a one-stop portal for all **SEMiOne** users. The modules in **PROSemi** are as follows:

- End-User Site
- Admin Site
- Backend Applications –
 - Certification Expiry email alert System
 - Release Pending Tickets

3.2. Features

SEMiOne designated users can access the End-User website and Admin website modules through the intranet. These two websites are password-protected. There would be access restrictions for various types of users to access different features of the sites. There should be two back-end applications - One is an alerting system to remind the users of their certification due dates.

End-User site

This site facilitates **SEMiOne** designated users to perform the following:

- Login into the **PROSemi** system
- To view their profile and to change the password
- Reserve resources in the form of tickets
- View all Trip tickets and also search/retrieve tickets based on the given search criteria
- Edit the Tickets Information
- Customer Pricing for Equipment and Human
- Generate Invoice

- Make Invoices
- Edit Invoiced Trip Ticket Information
- Transfer the Invoices Data to the SAP System.
- Synchronization of the PO Amount.
- View the History of the Ticket's Transactions
- Generate Reports

The above tasks are available to the user based on the user's privileged access rights which are assigned by the **PROSemi** administrator.

Admin site

This module should allow the administrator to create users, manage master data, and control the user access rights for **PROSemi**. The administrator(s) can also generate reports pertaining to Users & Audit actions. This facilitates the administrator(s) to view the history of changes being done for ticketing information to the **PROSemi** with the feature **Audit History**.

3.3. Assumptions and Dependencies

- **PROSemi** depends on the Purchase Order (PO) data to complete the reservation of resources.
- **PROSemi** expects PO data as a replicated table in the **PROSemi** database. The data in this table should be imported from the production database at regular intervals.
- **SEMiOne** LDAP is intended to be used as a source for usernames for the users. Limitations, if any, like the same source not being used by different external systems might result in not being able to resolve the username in the audit screens.

4. Functional Requirements

4.1. End user site

4.1.1 Login

REQ #1

An interface should be developed for the administrator to submit credentials and log into the site. It should contain an entry area for **Username**, **Password**, and **Login Type** (indicating **SEMiOne** User or Other User) with **Submit** and **Reset** functionality.

REQ #2

Screen should allow performing the following actions:

- **Submit:** Validates the data entered against the active directory (LDAP) or SQL database based on the login type selected by the user.
- **Reset:** Reset the details entered in the screen.

REQ #3

The details of the **Active Directory** to be used should be configured in the application settings file.

REQ #4

- Success Login, the user should be navigated to the View Tickets page.
- Failure Login, the user should be prompted with an appropriate message.

4.1.2 General Requirements

This section briefs on the common requirements across all the features at the end-user site.

REQ #5

All the actions on the features of **PROSemi** are available for the users based on their access permissions assigned by the administrator. User should be prompted with an appropriate message when the user does not have required access rights on any action or feature.

REQ #6

For the entire search output screens, details should be displayed in the grid and each record should be associated with the checkboxes. Users can either select one record or more than one to perform any actions on the desired record.

REQ# 7

Below of each grid to the left, four links should be provided. These links should be used for navigating to different pages in the grid. Clicking “**First**”, “**Previous**”, “**Next**”, and “**Last**” links on the grid data displayed, the user should be navigated to first set, previous set, next set, and last set of the data grid respectively. Based on the number of records retrieved these links should be automatically enabled/disabled.

REQ # 8

Tracking should be made available for **Audit**. Tracking should handle only the transactions involved for tickets. The **AUDITLOG** is for tracking user actions.

- Table name
- Operation
- Fields Affected
 - Old Value
- New Value
- Username
- Created By
- Created On

REQ #9

For the reports output, header data should be associated with “**Sorting**” on clicking the same.

REQ #10

The “**Delete**” action across the site should be updating the selected record to inactive status. Inactive records should not be shown to the user for any feature retrieval. There should not be any physical deletion.

4.1.3 My Profile

This feature should facilitate logged in user to view his/her profile details.

REQ #11

An interface should be developed which should allow logged in user to view his/her profile details. Details displayed in the view screen are **Employee ID, Employee Name, Division, Login Type, Date Hired, Manager, and Certification details**, etc.

REQ #12

A link **“Change Password”** should be provided which should facilitate the logged-in user to change his/her password. This should be enabled only when the logged-in user is not **SEMiOne** user.

REQ #13

The old password given by the user should match the password exists in the database. Similarly, the new password and confirm password provided by the user in the interface should be the same.

REQ #14

Action for the **“Save”** button on the screen should update the employee information in the database with the new password if the required validations are passed.

4.1.4 End-User Logout

This feature should allow the logged-in user to close the session.

REQ #15

Action on the **“Log Out”** link which would be appearing at the top right corner should allow the user to close the session and navigate out of the site.

4.2. Administrative site

Admin user site functional requirements are listed here with a unique identifier for each requirement.

4.2.1 Login**REQ #16**

An interface should be developed for the administrator to submit credentials and login to the site. It should contain entry area for **“Username”**, **“Password”** and **Login Type** with **Submit** and **Reset** functionality.

REQ #17

Screen should allow performing the following actions:

- **Submit:** Validates the data entered against the active directory (LDAP) or SQL database based on the login type selected by the user.
- **Reset:** Reset the details entered on the screen.

REQ #18

If the login type is “**SEMiOne**”, validation would be done against the active directory configured in the *appsettings* file.

REQ #19

For the login type “**Others**” validation should be done against the **EMPLOYEE** for the existence and active status checks.

REQ # 20

On successful validation, the logged in user should be navigated to the Master Menu which lists all the Master screens. By default, the admin user should be navigated to the **Employee** maintenance screen.

4.2.2 Master Menu**REQ #21**

An interface should be developed which contains the list of all **Master Data entry features, User Management, and Reports** listed left side as a left navigation bar.

REQ #22

Top and bottom are the static controls. Top bar should be a static control that contains **SEMiOne** logo towards left corner and with text **PROSemi**. Towards right corner it displays the name of the logged in user with date and time, below these “**My Profile**” and “**Logout**” links should be provided.

REQ# 23

The bottom bar of the page should be a static control that contains the copyright text.

4.2.3 My Profiles

This feature should facilitate logged in user to view his/her profile details.

REQ #24

An interface should be developed which should allow the logged in user to view his/her profile details. Details displayed in the view screen are **Employee ID, Employee Name, Division, Login Type, Date Hired, Manager, and Certification details**, etc.

REQ #25

A link “**Change Password**” should be provided which should facilitate the logged in user to change his/her password. This should be enabled only when the logged in user is not **SEMiOne** user.

REQ #26

The old password given by the user should match the password exists in the database. Similarly, the new password and confirm password provided by the user in the interface should be the same.

REQ #27

Action for the “**Save**” button on the screen should update the employee information in the database with the new password if the required validations are passed.

4.2.4 Master Data Entry

This feature should allow the administrator to maintain the master data for the **PROSemi**. The following requirements depicts the data maintained for the various master tables.

The following are the general requirements for all the master data screens:

REQ #28

Clicking “**First**”, “**Previous**”, “**Next**”, and “**Last**” links on the grid, user should be navigated to the first set, previous set, next set, and last set of the data grid respectively.

REQ #29

An entry should be logged with the following information captured for all the actions (insert, update & delete) into the table **AUDITLOG** for the user actions tracking.

- Table name
- Operation
- Fields Affected
- Old Value
- New Value
- Username
- Created By
- Created On

REQ# 30

Search output should be displayed in a grid. Each record in the grid should be associated with a checkbox. Action on **“Delete”** in the screen for the selected records should make them inactive by updating the database accordingly. This should be allowed only on the active records.

REQ# 31

In the edit screens of all the master data, the interface should provide a field to select **“In Active Date”** and ensure that the admin should select some date greater than or equal to the current date when the status selected is **“In Active”**.

4.2.4.1 Admin Logout

This feature should allow the administrator to close his/her session.

REQ #32

Action on the **“Log Out”** link which would be appearing at the top right corner should allow the user to close the session for the admin user.

4.2.5 Release Pending Tickets

This is a back-end continuous running application, and it should run based on the application status in the configuration file.

REQ # 33

The application should be executed for every configured interval of time.

REQ # 34

Reads the trip ticket data from the database whose status is **“Pending”** and **“Cancel; Not Billable”**. The status of these records should be updated to **“In-Active”**.

REQ # 35

The application should send an email to the administrator to update on the status of the application.

4.2.6 Software Interfaces**➤ Development**

- Microsoft .Net framework 2.0
- C# .Net 2.0
- ASP.NET 2.0
- IIS 5.1 server or higher
- Visual source safe 6.0

- SQL Server 2005
- XML
- XSLT
- Production
 - Microsoft .Net framework 2.0
 - C# .Net 2.0
 - ASP.NET 2.0
 - IIS 5.1 server or higher
 - Visual source safe 6.0
 - SQL Server 2005
 - XML
 - XSLT
- Connection to the external systems:
 - Access to the **SEMiOne** Active Directory
 - Connection to the Customer Purchase Order database
 - Access to the SMTP server

4.2.7 User Interfaces

4.2.7.1 General Screen Design Guidelines

- A design goal should be that unnecessary or redundant confirmation message boxes are not implemented in the new system.
- A design goal should be that the usability of the screens in the new system must be good. The number of mouse clicks needed to complete a certain task is a good example.
- Some fields need not always be mandatory.

4.2.7.2 Screen Layout

My Profile

Change Profile

Reserve Resources

View Trip Tickets

Search Trip Tickets

Start Date

Customer Company

Trip Ticket Status

End date

Resource ID

GO

Search

Reports

Logout

Reserve Use Of Resources

Trip Ticket #: 123

Category:

Customer Company:

Person Using:

Email: ashiky@dlvio.com

PO #: PO#-001-78

ISE Division: ISE-DVI-001

Billable? ☒ Yes ☐ No

Lot#

Project Code#:

Device:

Reason:

Resource Type:

Resource ID:

Resource Configuration:

Start Time: Tue 11/13/2007

00:00

No. Of Hours:

Create From Existing Tickets

Check Availability

X	S.No.	Resource Type	Configuration	Resource ID	Date of Reservation	Time of Reservation	#of hours
<input type="checkbox"/>	1	Human	Configuration-1	RES-1	MM-DD-YYYY	HH:MM:SS	5

Figure 3: Reserve Use of Resources

Customer Company

Trip Ticket Status

End date

Resource ID

GO

Search

Reports

Logout

1. Reservations made for a customer for a given duration & all resources.

Customer Company: Start Date: End Date:

Trip Ticket Status:

Go

2. Reservations made for a customer for a selected resource type & Duration

Customer Company: Start Date: End Date:

Trip Ticket Status: Resource Type:

Go

3. Pending Trip Tickets for a given customer and for a given or ALL POs

Customer Company: Start Date: End Date:

Trip Ticket Status: P O Number:

Go

S.No.	Trip Ticket #	Customer Company	Category	Resource Type	configuration	Resource Id	Date of Reservation	Time of Reservation	# of hours
1	Trip Tkt-1	ABC	ABC	Human	Configuration-1	RES-1	MM-DD-YYYY	HH:MM:SS	5
2	Trip Tkt-2	XYZ	XYZ	Equipment	Configuration-2	EOP-1	MM-DD-YYYY	HH:MM:SS	3
3	Trip Tkt-3	ABC	ABC	Human	Configuration-1	RES-2	MM-DD-YYYY	HH:MM:SS	5

Figure 4: Reports

Page 21 of 23

X	S.No.	Trip Ticket #	Customer Company	Resource Type	Configuration	Resource ID	Date of Reservation	Time of Reservation	#of hours
<input type="checkbox"/>	1	- Trip Tkt-1	ABC	Human	Configuration-1	RES-1	MM-DD-YYYY	HHMMSS	5
<input type="checkbox"/>	2	- Trip Tkt-2	XYZ	Equipment	Configuration-2	EQP-1	MM-DD-YYYY	HHMMSS	3
<input type="checkbox"/>	3	- Trip Tkt-3	PQR	Human	Configuration-1	RES-2	MM-DD-YYYY	HHMMSS	5
<input type="checkbox"/>	4	- Trip Tkt-4	MNC	Equipment	Configuration-2	EQP-2	MM-DD-YYYY	HHMMSS	6
<input type="checkbox"/>	5	+ Trip Tkt-5	QRT	Human	Configuration-1	RES-1	MM-DD-YYYY	HHMMSS	8

Figure 5: Search Tickets

4.2.8 Communication Interfaces

- Active directory communication through LDAP.
- Customer Purchase Order data retrieval through ADO.NET objects.
- Email server through SMTP protocol.
- Any third-party integration (in future) can be through .Net web service.

5. Non-functional / Other Requirements

5.1. Security Requirements

- The website can be accessed by the authenticated users only.
- Access to the operations for each webform should be based on the user's privileges defined by the Admin user.
- All changes to the data should be stored in the audit table for verification and tracking.
- Sections related to the confidential data should be granted access to the concerned departments only.
- The details of the database server and other systems should be configured in the web.config of the **PROSemi** application in encrypted form.

6. Acceptance Criteria

Every milestone in the project lifecycle should be reviewed by the client. Client feedback and acceptance is critical at all stages. The client must review and sign-off all the documents submitted by **the Service Provider**.

7. References

Document Index	Document Name
Xyz	Business Requirements document
243	Admin Website Prototype document
abc	End User Website Prototype Document
12345	ERWIN Diagram