#Company\_Logo#

**Security Guide for**

**App\_Name**

*User guide for all the roles in App\_Name*

*The guide provides functional description of features and actions available in generated application.*

Company\_Name

Date\_Create

**COPYRIGHT**

Security Guide for App\_Name (Date\_Create at Time\_Create)

The information provided in this document, including URL and other Internet Web site references is "as-is" and for informational purposes only. The information contained herein is subject to change without notice and is not warranted to be error-free. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious and for only illustration purposes. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any section of application App\_Name. You may copy and use this document for your internal, reference purposes.

© 2016 Company\_Name. All rights reserved.

# Document Revision

Table 1: Revision History

|  |  |  |
| --- | --- | --- |
| Version | Date Published | Key Notes |
| 1.0 | Date\_Create | First Release |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[Document Revision 3](#_Toc442711462)

[Introduction To Security 7](#_Toc442711463)

[Security is Critical 7](#_Toc442711464)

[Security Features in App\_Name 7](#_Toc442711465)

[Introduction 7](#_Toc442711466)

[Types of Security 7](#_Toc442711467)

[Types of Securities displayed in this Document 9](#_Toc442711468)

[Configurable Security Settings 9](#_Toc442711469)

[Unchangeable Settings 9](#_Toc442711470)

[Not Covered 10](#_Toc442711471)

[TERMINOLOGY 11](#_Toc442711472)

[APPLICATION USER 11](#_Toc442711473)

[ROLE 11](#_Toc442711474)

[PERMISSION 11](#_Toc442711475)

[ACTIONS 11](#_Toc442711476)

[ENTITIES 12](#_Toc442711477)

[PROPERTIES 12](#_Toc442711478)

[FIELDS 12](#_Toc442711479)

[BUSINESS RULES 12](#_Toc442711480)

[ATTRIBUTES 12](#_Toc442711481)

[VERBS 12](#_Toc442711482)

[SELF-SERVICE 12](#_Toc442711483)

[ROLE-LEVEL SECURITY 14](#_Toc442711484)

[ROLE-LEVEL SECURITY MODEL EXPLAINED 14](#_Toc442711485)

[User with Multiple Roles 15](#_Toc442711486)

[ROLE-LEVEL SECURITY Model in APP\_NAME 16](#_Toc442711487)

[ENTITY LEVEL SECURITY 17](#_Toc442711488)

[FIELD LEVEL SECURITY 17](#_Toc442711489)

[BUSINESS RULE BASED SECURITY 18](#_Toc442711490)

[TODO: Example—Asset Management Application 19](#_Toc442711491)

[DESCRIPTION 19](#_Toc442711492)

[Permission Settings for Roles 19](#_Toc442711493)

[Field Level Security (FLS) for Roles 21](#_Toc442711494)

[Assign Users for Roles 21](#_Toc442711495)

[Points To Remember 21](#_Toc442711496)

[Implementing Security from Admin Section 22](#_Toc442711497)

[Step 1: Creating Role in App\_Name 23](#_Toc442711498)

[Step 2: Permission Setting For Roles (Entity Level) 25](#_Toc442711499)

[Step 3: Set Field Level Security For Roles (FLS) 28](#_Toc442711500)

[Step 4: Assign Users to Role 29](#_Toc442711501)

[Create User 29](#_Toc442711502)

[Create User in Shared User System 30](#_Toc442711503)

[Method 1: Assign Roles to User 32](#_Toc442711504)

[Method 2: Assign Users to Role 32](#_Toc442711505)

[Dynamic Roles 33](#_Toc442711506)

[**Test It Out: Using Applications by User Login** 35](#_Toc442711507)

[Home Screen For Users 36](#_Toc442711508)

[User- Technical Supervisor 37](#_Toc442711509)

[User- Technical Manager 38](#_Toc442711510)

[User Based Security 39](#_Toc442711511)

[How User Based Security works? 39](#_Toc442711512)

[Application Configuration Settings 41](#_Toc442711513)

[Active Directory Settings 42](#_Toc442711514)

[Security Compliance Settings 42](#_Toc442711515)

[Other Settings 42](#_Toc442711516)

[Add Application Settings or Group 43](#_Toc442711517)

[Multi-Tenant Security 44](#_Toc442711518)

[Steps In Model Generator 45](#_Toc442711519)

[Test It Out 47](#_Toc442711520)

[Entity Visible 47](#_Toc442711521)

[USER DEFINED PAGES 49](#_Toc442711522)

[Creating User Defined Page in App\_Name 49](#_Toc442711523)

[Select Page 49](#_Toc442711524)

[Select Roles 49](#_Toc442711525)

[Custom Information 49](#_Toc442711526)

[TEST IT OUT 50](#_Toc442711527)

# Introduction To Security

## Security is Critical

With the increasing use of internet based applications for managing business data, it is important to protect the data from unauthorized use, disclosure, modification or destruction. Ensuring that users have the proper authority to see the critical information/data, load new data, or update existing data is an important aspect of application development. User errors, negligence, or attempted manipulation on your system should not result in loss of information or processing time. Do all users need the same level of access to the data and to the functions provided by your applications? Are there subsets of users that need access to privileged functions? Are some documents restricted to certain classes of users? The answers to questions like these help provide the basis for the security requirements for your application.

## Security Features in App\_Name

### Introduction

**App\_Name** include a powerful and flexible security model to protect your data according to your security requirements. The **App\_Name** application provides you with powerful but flexible and configurable security mechanisms, which enables you to set up application security with the level of granularity needed by your security requirements.

The goals of the model are as follows:

* Provide users with the access only to the appropriate levels of information that is required to do their jobs.
* Categorize users by role and restrict access based on those roles.
* Prevent a user's access to records the user does not own or share.
* Maintaining a high level of customer privacy
* Ensuring data security, protects business interest, and ensures compliance with government regulations.

### Types of Security

To understand the document and settings better, the table below gives a quick summary of the different security concepts supported by *App\_Name*, and their impact on application behavior.

|  |  |
| --- | --- |
| Name | Description |
| Roles | Roles typically represent a group of real life user roles that have a set of permissions. A user has to be in a role to get application access and can only perform the actions defined by the permissions granted to that role. |
| Permissions | Permissions are typically the CRUD (Create Read Update Delete) rights to entities for a given role. In addition, other actions (referred to as Verbs and Self Security) are also controlled through these permissions. |
| Field Level Security | You may need to give someone view and edit rights to a record, but not to all fields in a record. Field Level Security for a role allows you to have finer grain control over who can view and edit which fields in an entity. |
| User based security | With user based security setting users see only those records that are associated with their user id. They will thus not be able to see any records connected to other users.  This allows you to enable an end customer to login and see his orders. User based security can be ignored for certain roles, allowing members of that role to view records for everyone. |
| Self-Service | If self-service is enabled on an entity for a given role, then users in that role will be able to exercise the permissions granted (e.g. edit), but only on those records that they are connected to (similar to user based security). In this scenario, they may be granted rights to edit records, but the self-service will restrict the capability so they can edit only their own record. This is different from User based Security, which essentially converts the application into a multi-tenant application – and only the records for logged in user are returned from database.  (Self-service is not applicable for all the entities but the entities that are associated directly or indirectly with ‘user’ entity. |
| Auto-Registration | In application the new users can register themselves and create a record of their own. The user though would not have privilege to see any of the application content unless provided with a role either through admin, default role for new users or dynamic role mapping.  In addition, if a record for the user does not exist (e.g. a Member entity record for a new user that just registered would not exist), then the user is allowed to create that record and connect his user id to that record. |
| Multi-Tenant Security | This is implemented at the code level and cannot be modified by the end administrators. Currently, this application (does NOT have / has) multi-tenant security (based on the entity TODO: <Entity Name>).  Thus a user can see only the records that are connected to the organization or entity that he is associated with. And all the roles or permissions that are then defined are defined within that context.  Essentially, this setup disallows multiple organizations to see each other’s data, even though they are hosted on the same system. |
| Login Attempt Control | The administrators of application can control the attempts to login into application by forcing password change, no. of failed login and locking the users from accessing the application. |

In addition, there are Audit Trails which track changes and deletions made to your data, along with the user who made the change.

Note: The security features discussed here are all-in-one for a standard application. Based on settings some security features may not be available in an application.

## Types of Securities displayed in this Document

The document covers security in App\_Name in these sections:

### Configurable Security Settings

1. **Role Based Security View:** The section takes one role at a time and displays the tables of entity level permissions and Field Level Security for it.
2. **Dynamic Role:** This section will cover the dynamic roles – where a user can be placed into a certain role dynamically based on the value of a certain property within an entity (e.g. Employee titled “Major” will automatically get the privileges of “Major Role” or a role that is mapped with the dynamic value of “Major”).
3. **Entity Based Security View:** The section displays the inverse view of the above – for each entity, it describes the different roles that can act on it.
4. **User based Security:** All records in an entity can be restricted to only those that are somehow related to the logged in user. This section displays all those entities that are restricted in this manner.
5. **Hierarchical Security:** This allows an entity protected by user based security to be acted upon by hierarchical supervisor.
6. **Login Security Compliance:** This allows administrators to control the login behavior of users by locking users after specific no. of failed attempts and forcing password change after particular no. of days (for example 60 days). The user once locked will be unlocked automatically on completion of lock duration or can be unlocked earlier by admin only.
7. **Custom Role Based Pages:** This section describes which roles have specific custom pages defined. This enables users of that role to see a totally different page than the default one.

### Unchangeable Settings

1. **Multi-Tenant Settings:** An application can support multiple organizations within it, while separating each organization data right at the base code and database level.
2. **Role Based Workflow Settings:** The actions that a user can take on a workflow are defined in the application model and are NOT configurable by an administrator.

### Not Covered

1. **Conditional Security View:** The application may choose to hide or show or make certain fields un-editable based on business rule and other settings. These conditional security settings are NOT covered in this document and can be found in the business rules document.
2. **Users in a Role:** While it is possible to display the different users that are members of different roles, this list is not included in this document.

# TERMINOLOGY

This section defines the following terms, which are used throughout the security documentation:

## APPLICATION USER

Application users are the individuals who can access the application. The person can become an application user through registration process either by himself or application admin. The application user (if not admin) has rights to create update delete and view (CRUD) the records in application based on role assigned to him.

**The users of application can be locked or unlocked (other than login security compliance) directly by admin from user list. If a user is locked through login security compliance, it can be unlocked only by admin before the lock period is expired.**

## ROLE

Roles determine the features and tasks users can access. Roles are the fundamental building blocks that you use to implement your security. One role can be specified to one or more users. A single user can be granted several roles, each of which separately determines access rights to the entities or features, the capability to perform certain actions. For example, a user can have the Sales Manager role in addition to being a Customer Service Representative, in which case that user has all the access permissions of both roles. Admin role is capable to manage the complete application including but not limited to:

* Managing users and their permissions,
* Configuring all organization-wide settings,
* Deactivating users, viewing the login history and
* Exercise CRUD operations for all data.

## PERMISSION

Permissions are security characteristics of documents that associate a role with a capability to perform following actions in respective features and tasks:

* View
* Add
* Edit
* Delete
* Disabled verbs
* Self-Service

## ACTIONS

Permissions dictate what actions application users can and cannot perform in the role. Following actions or combination of actions can be performed:

* View - permits the user to view a record (read-only access)
* Add - permits the user to add a record
* Edit - permits the user to edit an existing record
* Delete - permits the user to delete a record

## ENTITIES

Entities are the self-existing independent features of application. All the tasks and features of application are broadly classified as entity.

## PROPERTIES

Properties are the attributes that characterize an entity. Every entity you create has ‘Name’ and ‘Description’ properties by default. You can add any additional attributes you want to tell about an entity as property.

## FIELDS

Fields are the placeholders where the data is saved. The fields can be at any level i.e. application, entity, property or record.

## BUSINESS RULES

Business rules are the custom ‘condition’ and ‘action’ arguments. The business rules are applicable at property level and if a condition is provided to a property than respective action will be taken at same and parallel properties.

## ATTRIBUTES

Attributes are the properties of a subject that are used to describe a specific thing and does not have any reference or coordination with properties of App\_Name application.

## VERBS

The user created verbs for an entity(s) allows actions other than CRUD actions. The verbs are by default enabled for a role but can be disabled using permission settings.

## SELF-SERVICE

With self-service enabled, users have access to all the records in an entity but the ability to update or delete them. The complete CRUD permissions are available only for the records owned or created by user.

# ROLE-LEVEL SECURITY

This section describes the role-level security model implementation in App\_Name.

## ROLE-LEVEL SECURITY MODEL EXPLAINED

Role-Level Security enables access to entities/features and data through user roles rather than only users. User access is based on the definition of the roles assigned to the user. Role-level security secures access in a "Who can see what and do what on which entities/features or data under which conditions" approach. Here “Who” is the application user?

Role-Level security is built on the premise that users are authenticated, which is the process of identifying the user. Once identified, roles and permissions are assigned. This is accomplished with a one-time setup, in which permissions, responsibilities, and other roles are assigned to the role. Users are not required to be assigned the lower-level permissions directly, since permissions are implicitly inherited on the basis of the roles assigned to the user. This simplifies mass updates of user permissions, since an organization need only change the permissions or role inheritance hierarchy defined for a given role, and the users assigned that role will inherit the new set of permissions automatically.

Organizations can define roles that closely mirror their business situation. The following conceptual diagram shows how permissions are assigned to roles and roles assigned to application user:

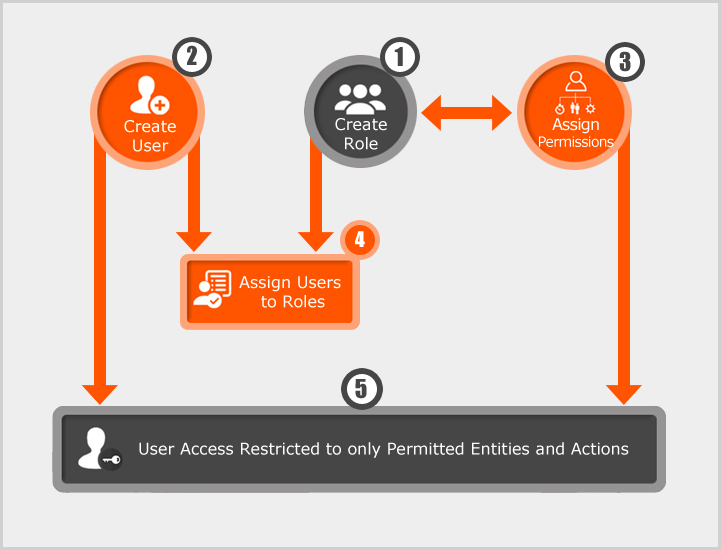


Figure 1: Role Level Security Model

### User with Multiple Roles

Furthermore, individual users (e.g. John, Mary) might be assigned to one or more roles, where the roles are based on the user's job responsibilities and competencies in the organization, none of which are defined by a single role.

If you assign multiple roles with conflicting permissions, then the user is granted the more permissive permission, which is the union of all the permissions granted to the multiple roles. For example, if you assign to the same user a role that allows creating a snapshot and a role that restricts it, then the user is allowed to create snapshots.

EXAMPLE OF USERS WITH MULTIPLE ROLES

As shown in Figure 2, John Smith and Mary Walters are assigned with two roles in application:

* John Smith as a member assigned **EngHead** Role has View, Add, Edit and Delete permissions to Engineering Documents entity. We do not want him to view purchase docs.
* Mary Walters as a member assigned **HRHead** Role has View, Add, Edit and Delete permissions to Human Resources Documents entity.
* Jane Doe is a member assigned with two roles **EngHead** and **Purchase User**. As a member of EngHead role she has all related permissions and as purchase user has permission to view (only) purchase docs.

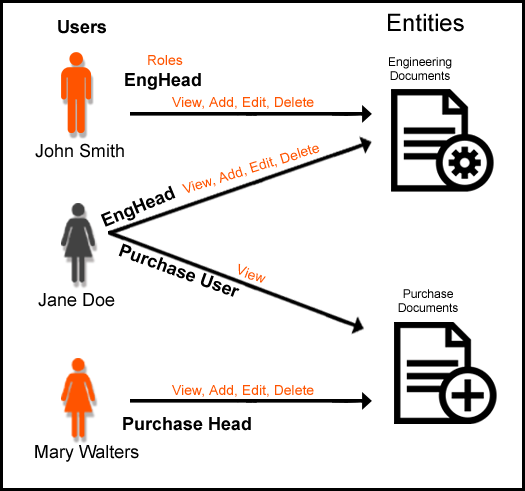


Figure 2: Multiple roles assigned to user

## ROLE-LEVEL SECURITY Model in APP\_NAME

This section describes the Role-Level Security implementation in **App\_Name Application.** The role-level security is in the form of three levels as shown in Figure 3:

1. ENTITY LEVEL SECURITY
2. FIELD LEVEL SECURITY
3. BUSINESS RULE BASED SECURITY

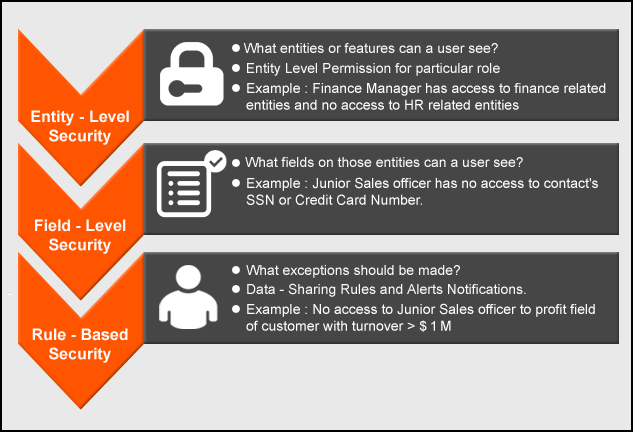


Figure 3: Role-level Security levels

### ENTITY LEVEL SECURITY

This is the simplest and most common approach to defining role-level security. With this approach, access to all attributes of a specific entity or feature is defined for each user role. If the user does not have a role that is entitled to access the attribute then it will be restricted.

### FIELD LEVEL SECURITY

So far we reviewed application role from an entity perspective, but what if there are sensitive data fields on the entity record that you want to secure? We refer to managing the securing to and updating data at a more granular level (individual field level) as Field Level Security.

The field-level security would provide the ability to:

* Hide certain fields from particular groups of users
* Allow particular groups of users to view but not edit certain fields of an entity

Some examples of data fields you want to secure:

* A contact’s SSN or credit card numbers
* An Account’s sales numbers
* Home Phone Numbers

### BUSINESS RULE BASED SECURITY

Some real-life scenarios require a more complex access strategy or policy, which can involve specifying a series of rules to define the conditions under which an entity’s attributes can be accessed.

Example of BUSINESS RULE:

Consider a scenario in which business rules require that all records with a customer turnover:

* Greater than or equal to $1M have the profit field restricted or hidden
* Less than $1M have the profit field displayed

The App\_Name Application business rule settings can be configured to filter out and display records after applying the business rule.

In addition business rules can be set for conditions and trigger actions as configured like E-Mail Notifications, Alerts. For business rules see the business rules guide for App\_Name.

# TODO: Example—Asset Management Application

This section describes a scenario that uses Role Level Security. The scenario is not meant to demonstrate the correct way to set up application security, as your situation is likely to be unique. However, it demonstrates how role level security works and may give you the ideas for how to implement your own security model.

## DESCRIPTION

TODO: Write example of your application

The example application: Asset management is created in using Model Generator. The application is a pseudo framework and created with sole purpose to guide users of App\_Name. For the Asset Management application used by a company, the various features or modules have restricted access based on the department or job function or responsibility of the employee. Additionally, some data or fields can only be accessed by employees with certain job functions. This guide demonstrates the roles and security taking the example of this application.

For simplicity we have selected two Roles and Users for explanation in below section. You may create many roles that closely mirror their business situation e.g. Visitor, Sales Supervisor , Sales Manager, Department Head, Human Resources, Division Head

To run through the example, perform the steps in each of the following sections:

* Create Roles
* Permission Setting for Role (Entity and field level)
* Create User
* Map Users and Roles
* Test the Settings
* Business Rules to Roles
* Test Business Rules Settings

### Permission Settings for Roles

For sake of simplicity only two roles ‘Technical Supervisor’ and ‘Technical Manager’ are described in following sections. The other roles can be created using similar manner.

Table 2: Permission List of Role ‘Technical Supervisor’

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entity Name | Add | Edit | Delete | View | Disabled Verbs | Self-Service | Auto-Registration |
| Assets |  |  |  | ✓ |  |  |  |
| Item | ✓ | ✓ |  | ✓ |  |  |  |
| Job Plan |  |  |  | ✓ |  |  |  |
| Location |  |  |  | ✓ |  |  |  |
| Person |  |  |  | ✓ |  |  | ✓ |
| Preventive Maintenance | ✓ | ✓ |  | ✓ | Bulk Update  Bulk Delete |  |  |
| Preventive Maintenance Status |  |  |  | ✓ | Bulk Update |  |  |
| Preventive Maintenance Type |  |  |  | ✓ | Bulk Update |  |  |
| Site |  |  |  | ✓ |  |  |  |
| Work Order |  |  |  | ✓ |  |  |  |

Table 3: Permission List of Role’ ‘Technical Manager’

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entity Name | Add | Edit | Delete | View | Disabled Verbs | Self-Service | Auto-Registration |
| Inventory | ✓ | ✓ |  | ✓ |  |  |  |
| Inventory Status |  |  |  | ✓ |  |  |  |
| Inventory Type |  |  |  | ✓ |  |  |  |
| Store Room |  |  |  | ✓ |  |  |  |
| Assets |  |  |  | ✓ |  |  |  |
| Item |  | ✓ | ✓ | ✓ |  |  |  |
| Item Status |  |  |  | ✓ |  |  |  |
| Item Type |  |  |  | ✓ |  |  |  |
| Job Plan |  |  |  | ✓ | Bulk Delete |  |  |
| Location |  |  |  | ✓ |  |  |  |
| Person |  |  |  | ✓ | Bulk Delete |  | ✓ |
| Preventive Maintenance |  | ✓ | ✓ | ✓ |  |  |  |
| Preventive Maintenance Status |  | ✓ | ✓ | ✓ |  |  |  |
| Preventive Maintenance Type |  | ✓ | ✓ | ✓ |  |  |  |
| Site |  |  |  | ✓ |  |  |  |
| Work Order |  |  |  | ✓ |  |  |  |

### Field Level Security (FLS) for Roles

FLS for role ‘Technical Supervisor’ is as follows:

Table 4: Field Level Security for Technical Supervisor

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | Property Name | Cannot Edit | Cannot View |
| Item | Tax Exempt | ✓ | ✓ |

### Assign Users for Roles

Assign the users to role. To set this example, a fake name list (Table 5) is used.

Table 5: Users and Roles Assigned in this example application

|  |  |  |
| --- | --- | --- |
| **User Name** | **User ID** | **Role Assigned** |
| Weldon Meighan | Weldon\_Visitor | Visitor |
| Gary Howland | Gary\_techsup | Technical Supervisor |
| Mary Parker | Mary\_techmgr | Technical Manager |
| Carolyn Davis | Carolyn\_salsup | Sales Supervisor |
| Mary Oh | Mary\_salman | Sales Manager |
| Richard Hawkins | Richard\_dephead | Department Head |
| Myles Hall | Myles\_humres | Human Resources |
| Lyle Taylor | Lyle\_addinf | Additional Information |
| Carl Sommers | Carl\_divhead | Division Head |
| Ashley Thompson | Web.admin | Admin |

### Points To Remember

1. The App\_Name has the basic non-editable ‘Admin’ role. This admin is a super user with all the rights to operate the application. Admin creates roles and permissions for other users. Check if this sufficient for you or else
2. Create another role using Admin Option on Home Page and
3. Assign permissions to this role: View, Add, Edit, Delete, Disable Verbs and Self-Service.
4. Add Users to this role
5. Give View permissions to entities that are in dropdown
6. At the end, assign at least one user to every role to see the effects
7. The roles and names here are for illustrative purpose only. Actual role may vary according to application needs.

# Implementing Security from Admin Section

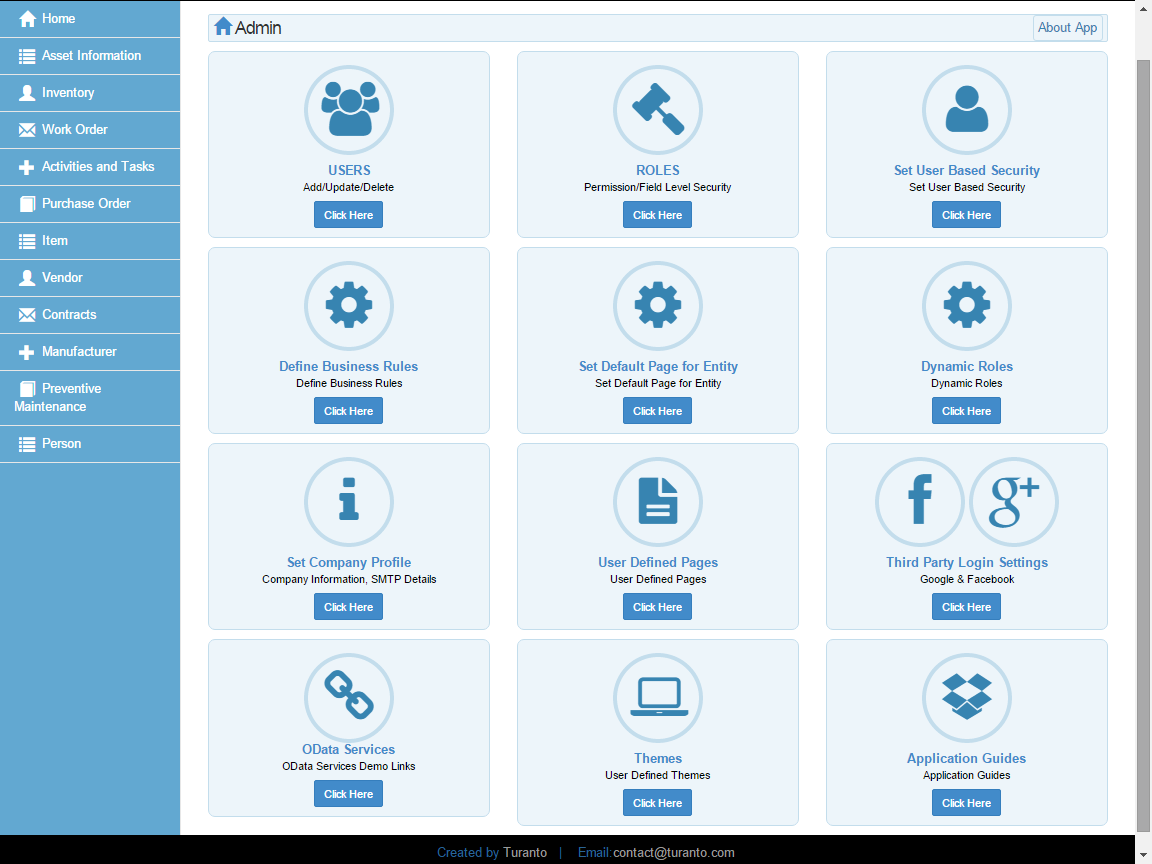
To run through the example, perform the steps in each of the following sections:

* Creating Role in App\_Name
* Permission Setting For Roles (Entity Level)
* Set Field Level Security for Roles (FLS)
* Assign Users to Role

This section includes:

* Defining the roles as required by application
* Providing permissions to roles
* Set Field Level Security to roles
* Assign Users to Role
* Test It Out (Screenshots from perspective of user)

To set up the application security for this scenario, you create the necessary roles, users with the example permissions. You will need admin access to the Application.



TODO: Replace this Image

Figure 4: Application Settings Page in App\_Name Application

The Roles and Permission Settings can be set in Admin Section.

Users: To view existing users and create new users

Roles: To add Roles, apply entity level and field level permissions

Set User Base Security: To apply user level security

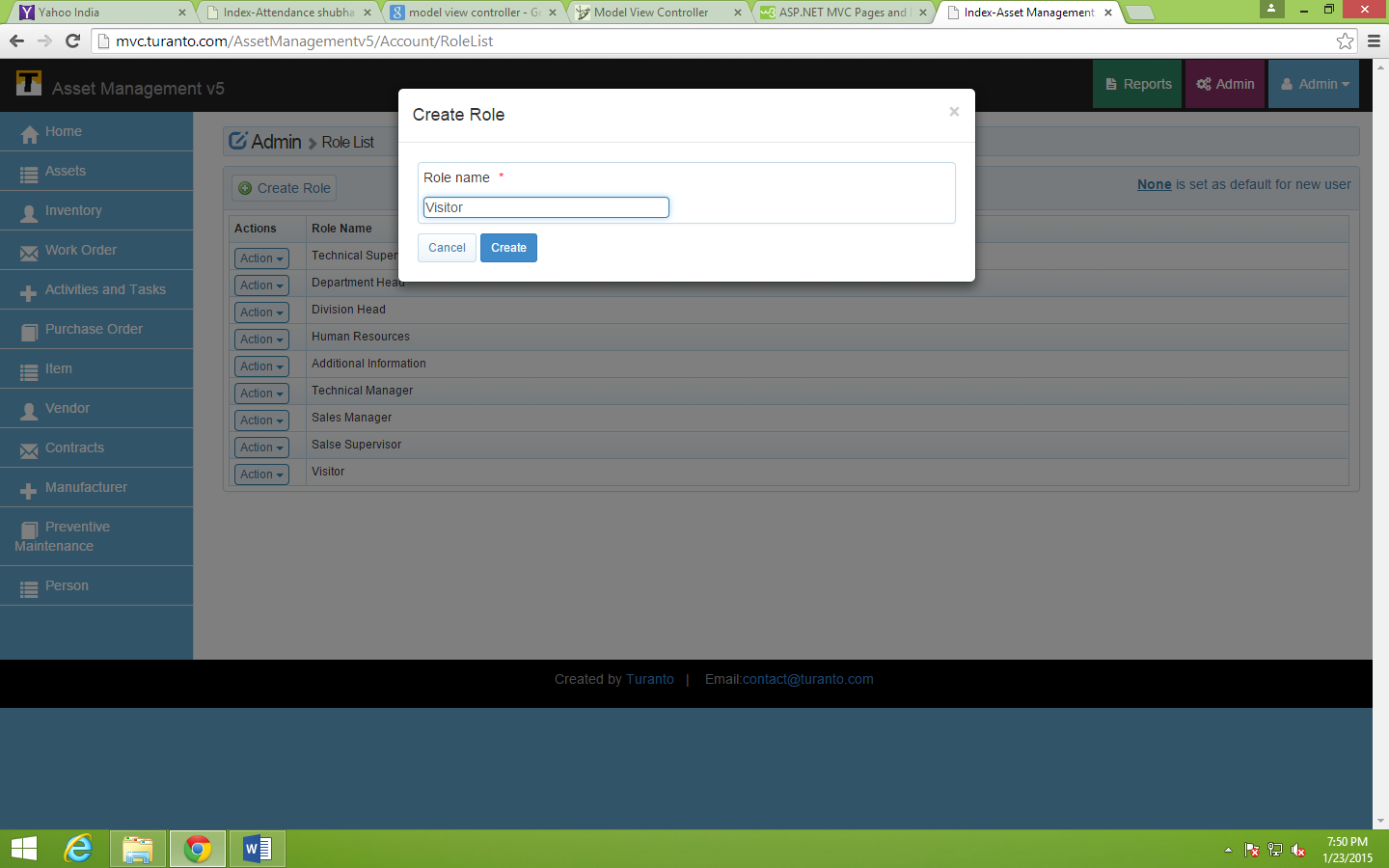
Define Business Rules: To create ‘actions’ for ‘case’ statements

Dynamic Roles: To assign a role based on the value of property (attribute)

User Defined Page: To create specific pages for every role.

## Step 1: Creating Role in App\_Name

You can create a new role from the ‘Create Role’ button in roles page (see Figure 6). This button opens a popup menu and asks for Role Name.



TODO: Replace this Image

Figure 5: Create a New Role

Role name: The role name is used to define a role. This field is mandatory and should be **unique**. If a role name is already present in role list (Figure 6), system will give an error.

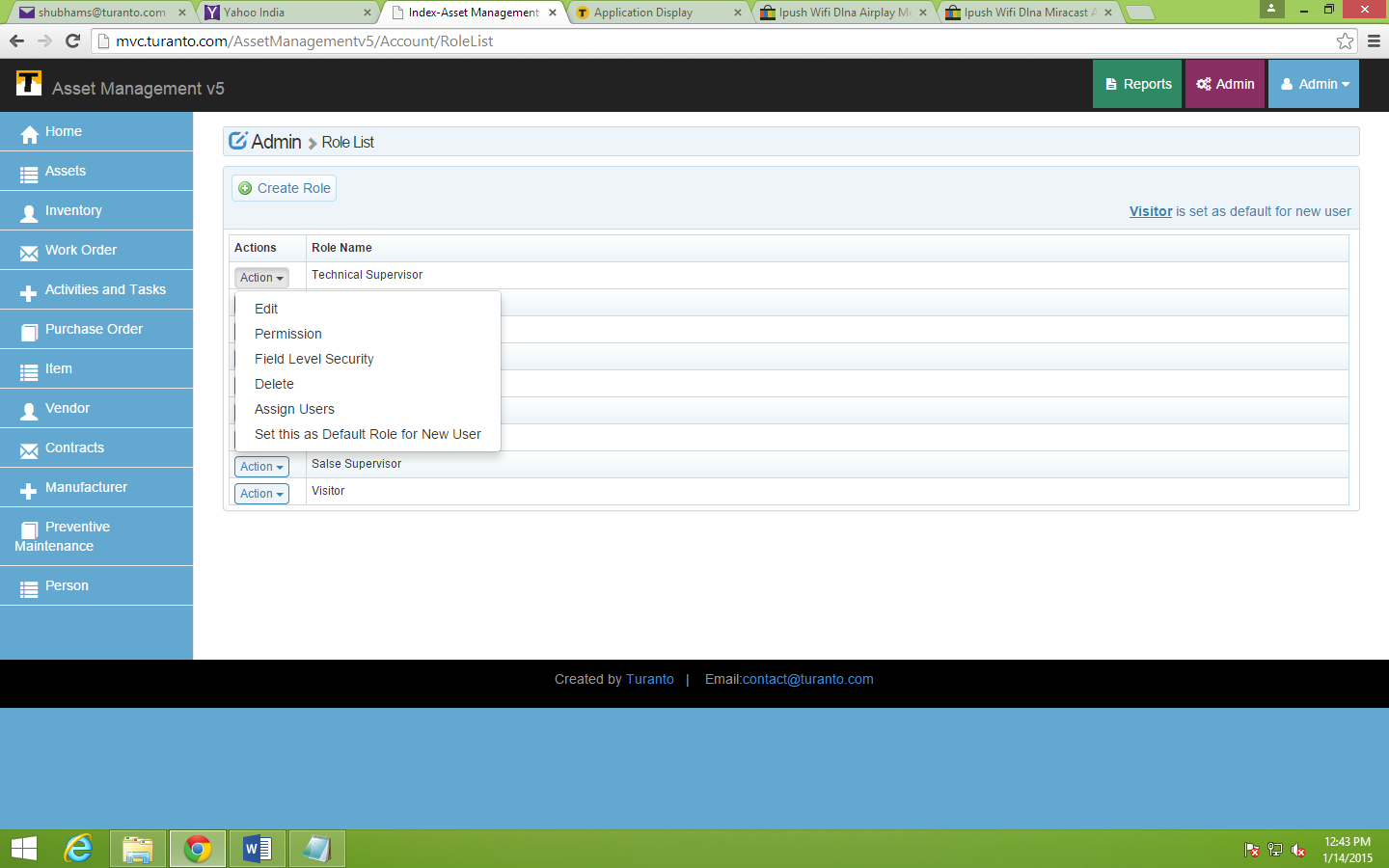


TODO: Replace this Image

Figure 6: Roles List

## Step 2: Permission Setting For Roles (Entity Level)

Entity level permissions provide view, add, edit and delete options at entity level. If a user is allowed to view the entity, he/she can see all the records of that particular entity. The user will not be able to see the items of dropdown if the user is not given the view permission in associated entity. The examples below list of all the permissions for respective roles entity vise.



TODO: Replace this Image

Figure 7: Actions List Associated with Role

Action- Edit: Edit option redirects you to the page where you can edit the name of role. This option is for entity level permissions in application.

Action- Permission: Permissions the action of officially allowing someone to do a particular thing; consent or authorization. In this option, permissions are applicable at entity level. Six types of permissions (view, add, edit, delete, disable verbs and self-service) are available in App\_Name. If a type of permission is selected in an entity, the role has accessibility to do particular operation in that entity.

**Self Service Only: Self Service option allows users to edit and delete only their own records i.e. records associated with their user id (or records associated to them) even after having create read update and delete permissions as defined in role permission settings. For rest of records they have only view permission. Self-service option is available on entities that are directly or indirectly associated with user id. The dropdown in self-association shows the name of association/entity by which the user id is governed.**

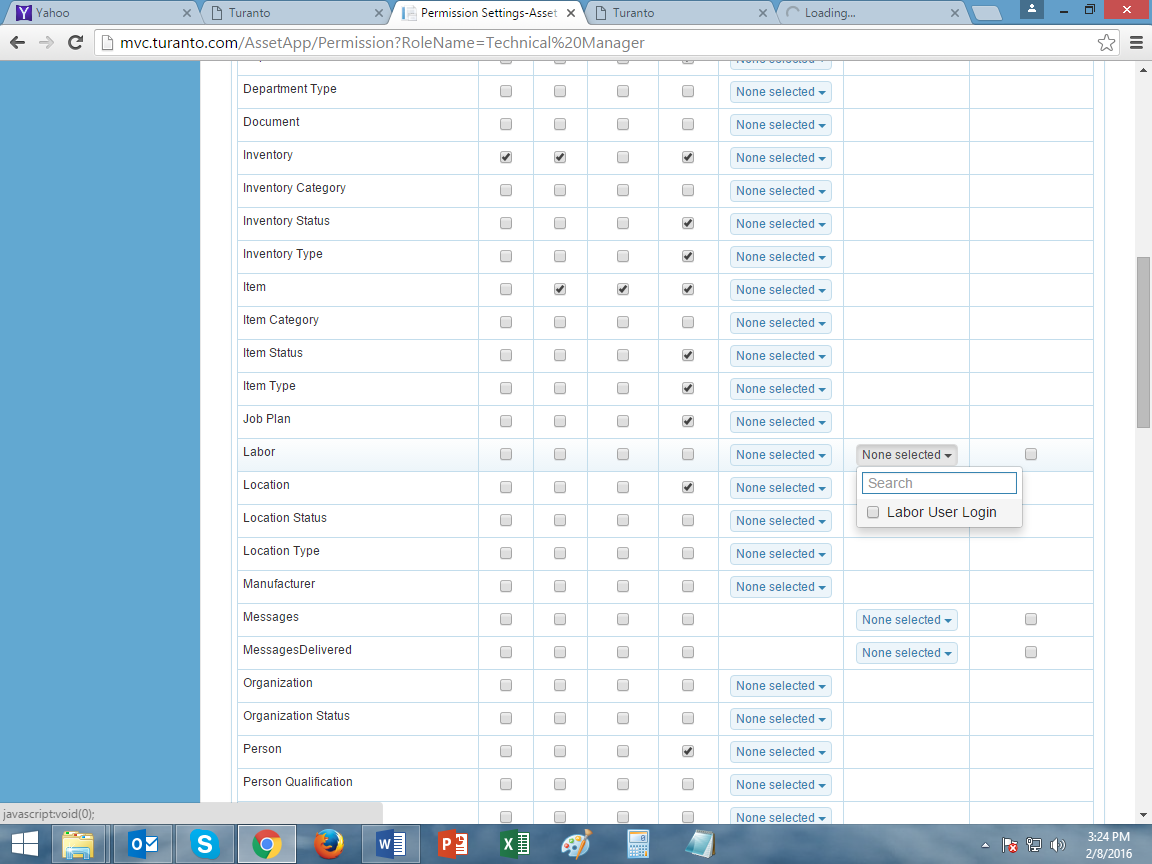


Figure 8: Self Service in Entities Associated with User

**Disable Verbs: Verb security option disables the verbs of an entity for the given role. The user with this role will not be able to see or edit the verbs. And also, the action or validation code set against these verbs will not be considered by application.**

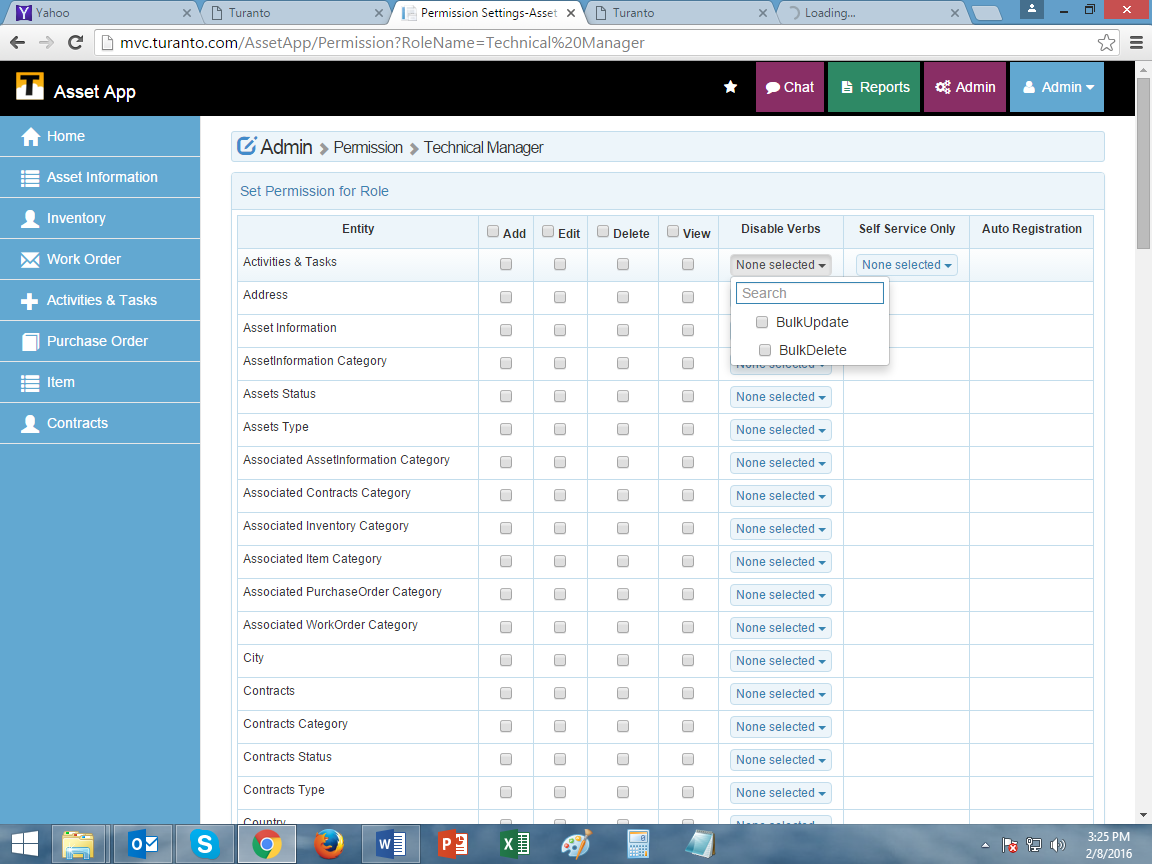


Figure 9: Permissions on Action Verbs

**Auto Registration: With user registration administrators can allow new users to create a record in entity (which is directly associated with user id) even if he is not assigned with any role and its concerned permissions. Auto-registration works with default role i.e. the role in which Auto-registration is activated should be set as default role for new users. In case if two roles have auto-registrations checked true, the role set as default will be in action.**

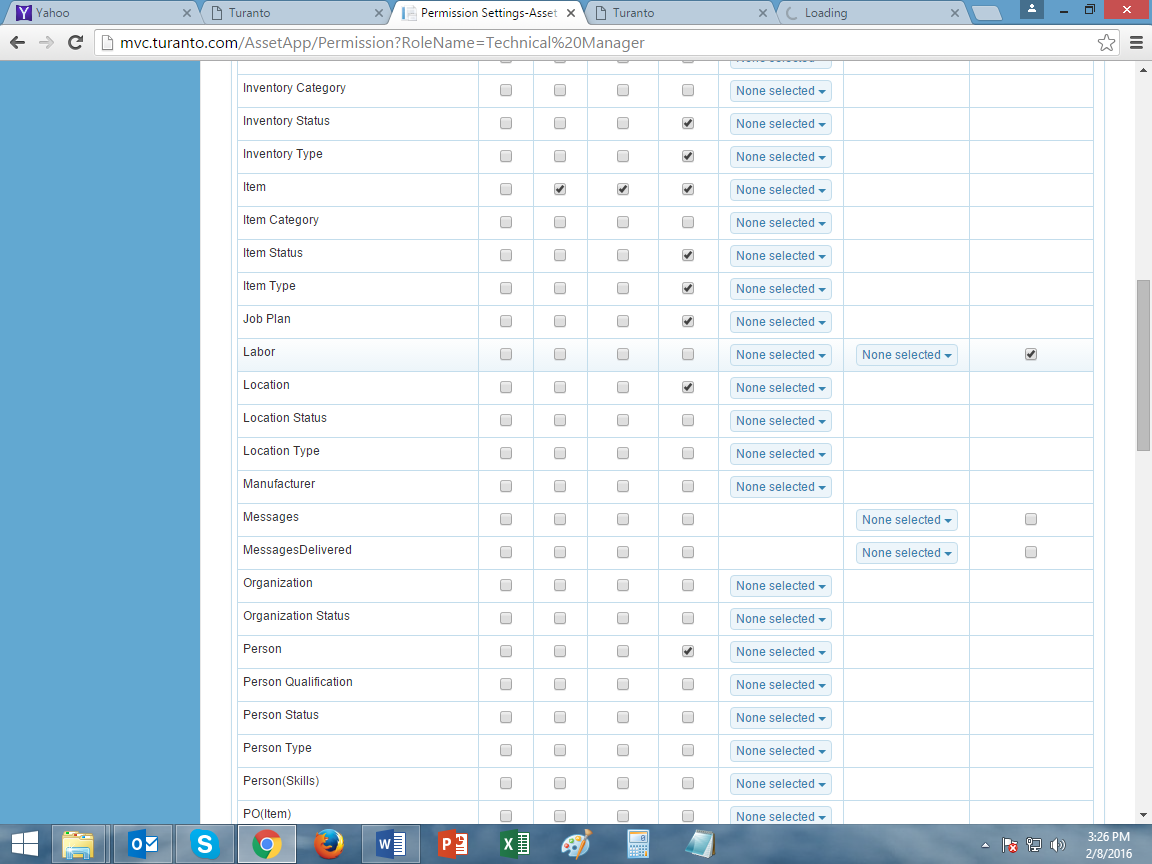


Figure 10: Permissions for Auto-Registration

To apply Entity Level Permissions in App\_Name

1. Click ‘Action’ on selected role (See Figure 7)
2. Select ‘Permissions’

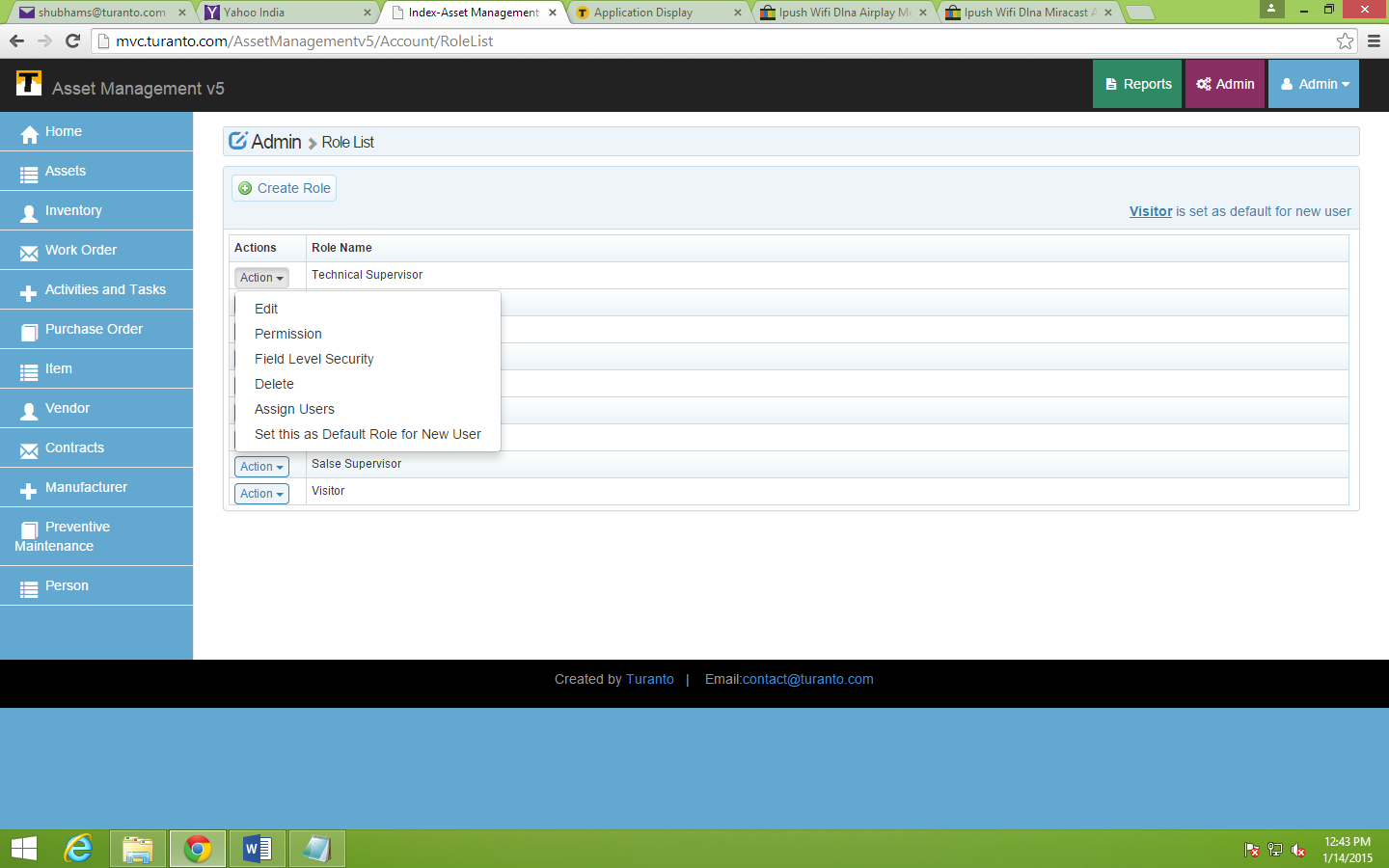


Figure 11: Assign Entity Level Permissions

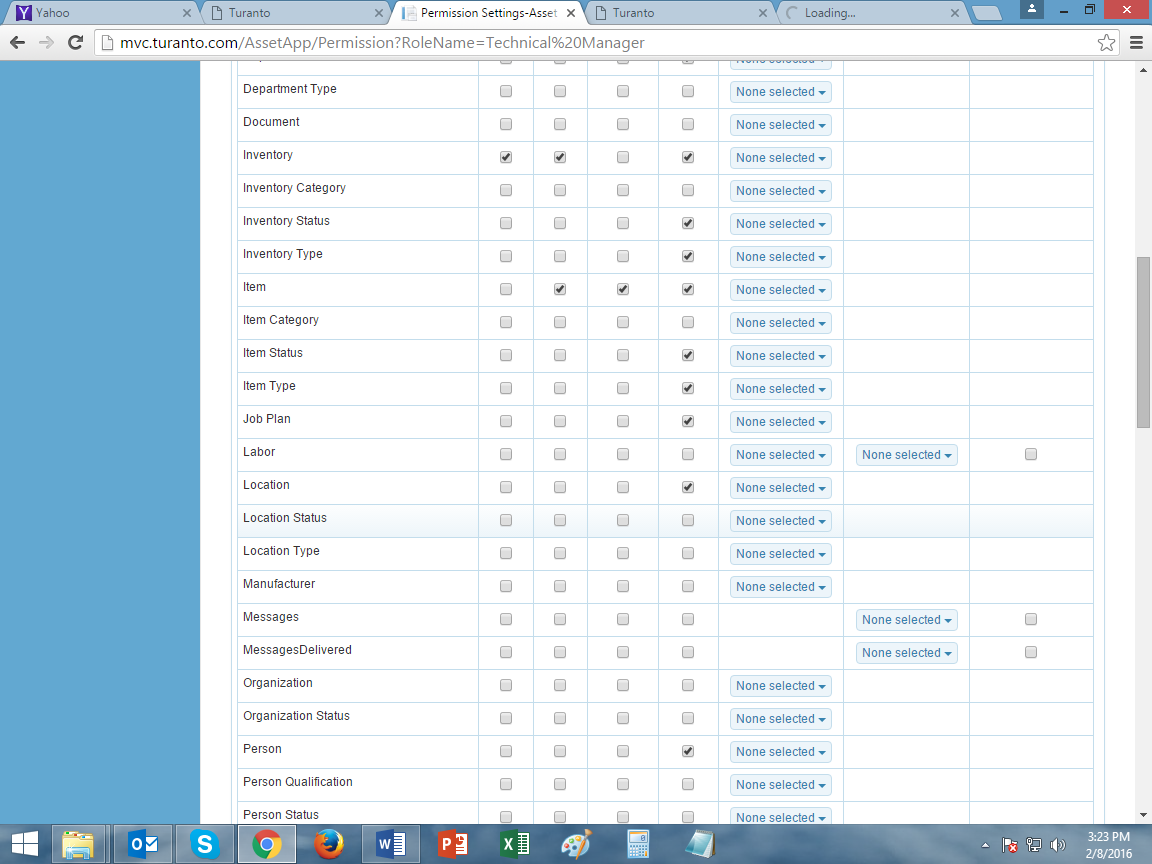
Screenshot of Entity level permissions for role: Technical Supervisor



TODO: Replace this Image

Figure 12: Role Permissions for Role- ‘Technical Supervisor’

Screenshot of Entity level permissions for role: Technical Manager



TODO: Replace this Image

Figure 13: Permission List for Role- ‘Technical Manager’

## Step 3: Set Field Level Security For Roles (FLS)

Field Level Security restricts some properties of given entity. The priority of field level security is higher than entity level permissions. The permissions in above section can also be defined from here. Right hand side displays the list of properties with selectable permissions of ‘Cannot Edit’ and ‘Cannot View’. By default, the role is allowed to edit and view properties of entity (if entity permission is enabled).

To apply Field Level Security in App\_Name:

1. Click ‘Actions’ on selected Role (See Figure 7)
2. Select ‘Field Level Security’

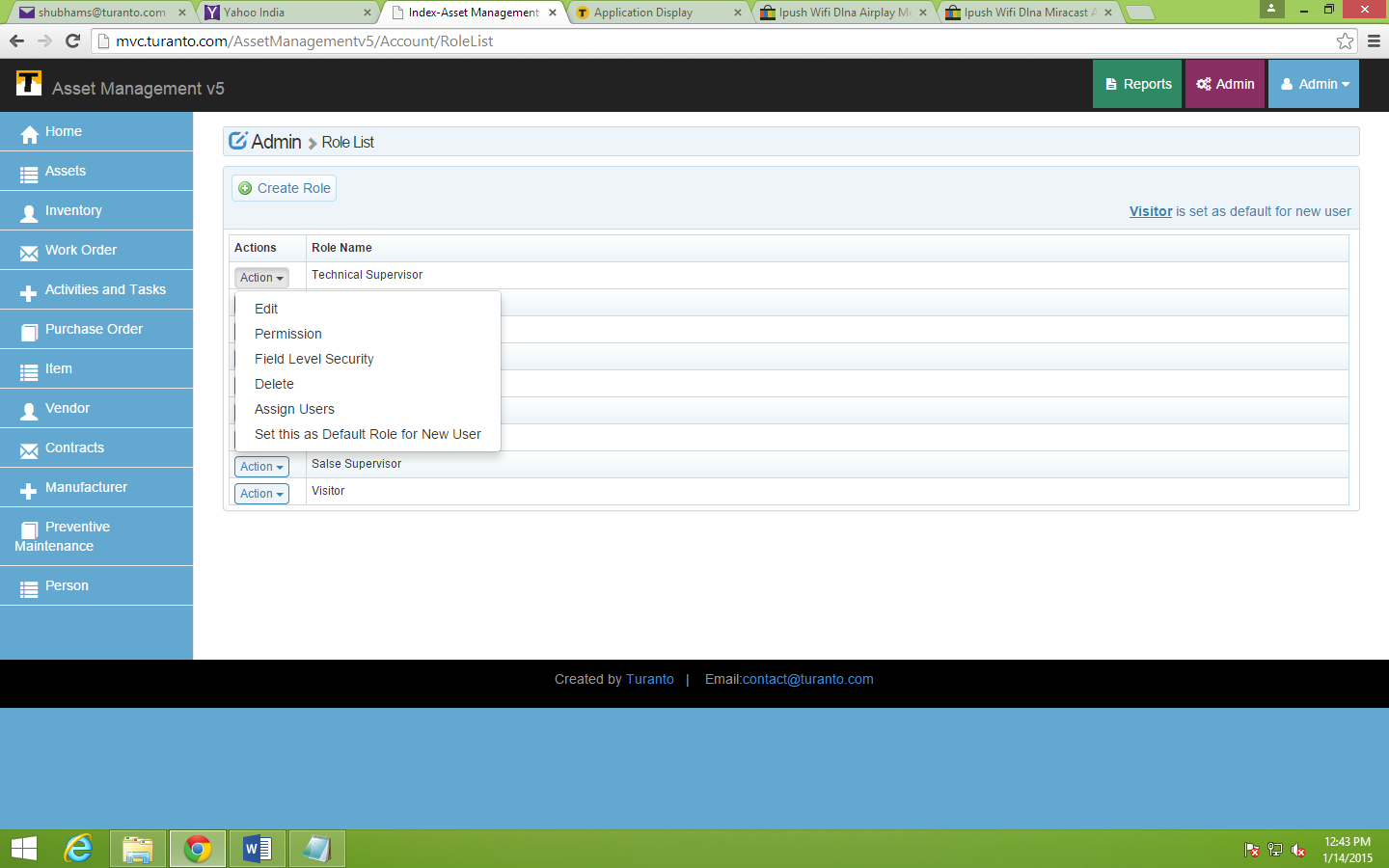
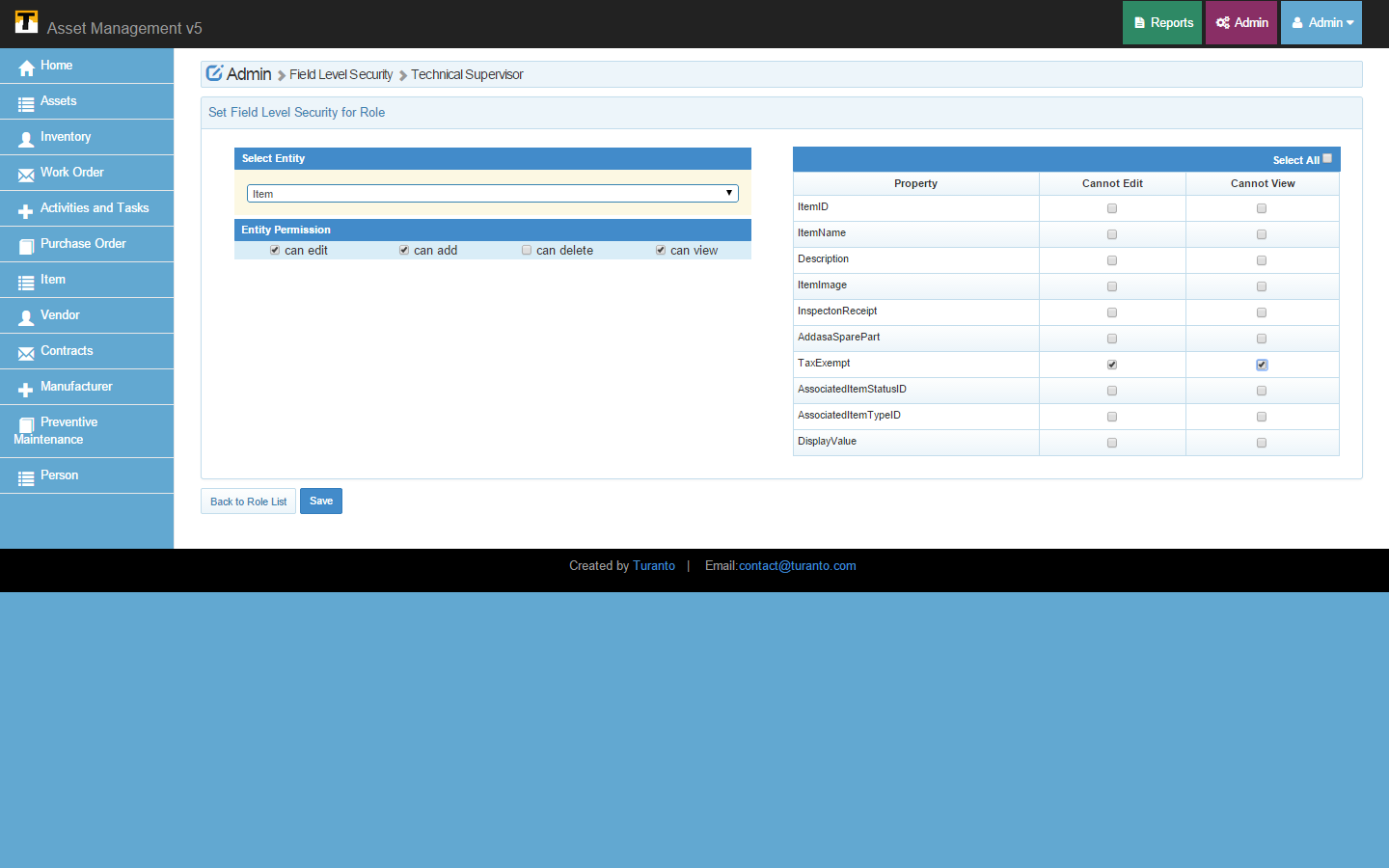


Figure 14: Selecting Field Level Security for Role

For example in ‘Item’ entity, Technical Supervisor cannot edit and cannot view ‘Tax Exempt’ property.



TODO: Replace this Image

Figure 15: Field Level Security

## Step 4: Assign Users to Role

The relation of roles and users can be defined either by assigning roles to a user or by assigning users to a role. Before this step let’s refer the step to create a new user from admin’s perspective.

### Create User

Users can be created using the Register on Login page of application. The admin can create a user using ‘Create User’. Figure 16 shows the Popup page displaying the fields to create a user. All these fields are mandatory.

User name: The username is used for login. This field is mandatory and should be unique. If a username is already present in user list (Figure 17), system will give an error ⚫Name visitor is already taken.

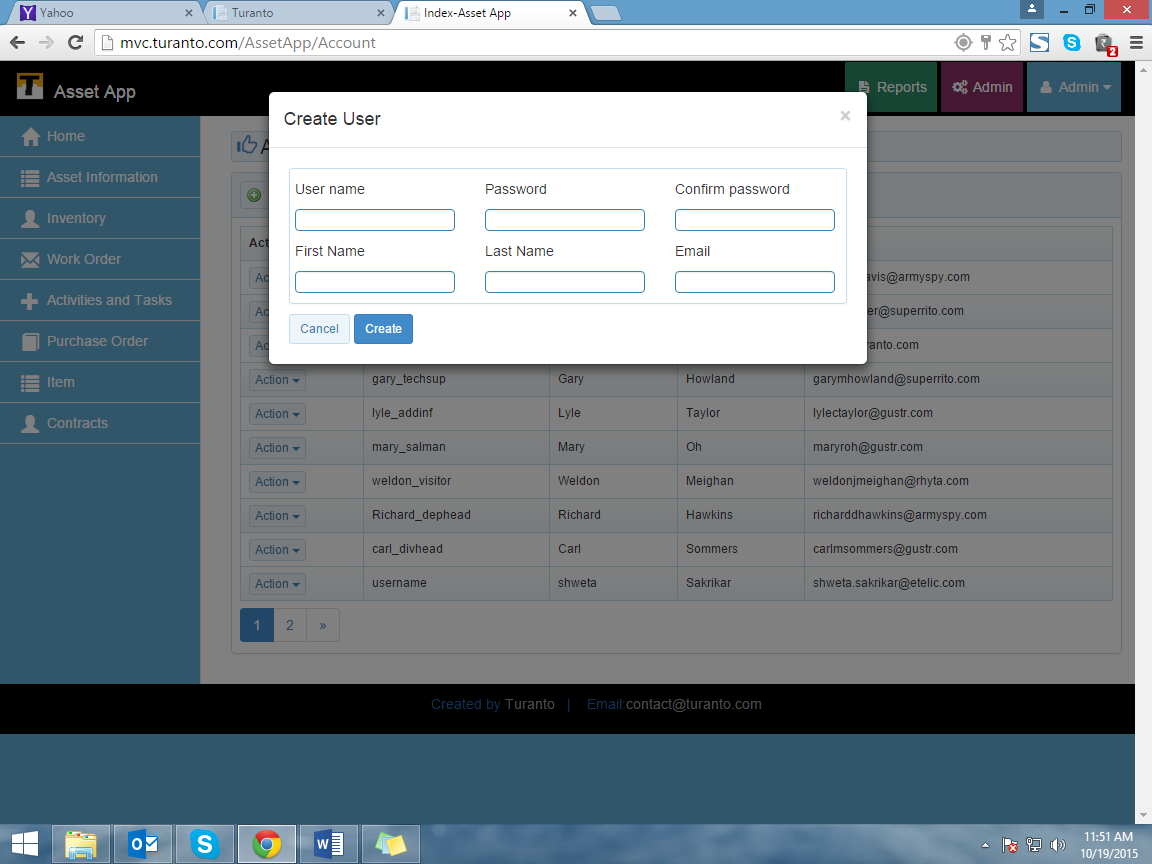
Password: The password should be at least 6 characters long and is case sensitive. It is advisable to keep a strong password (having an upper case, a number and a special symbol). You can only see bullets as you type your password.

Confirm Password: Retype your password here. This password should be exactly same as ‘password’ in above point.

First Name: Provide the first name of user.

Last Name: Provide the last name of user.

Email: Provide a valid email for future correspondence, workflow notifications and reports.

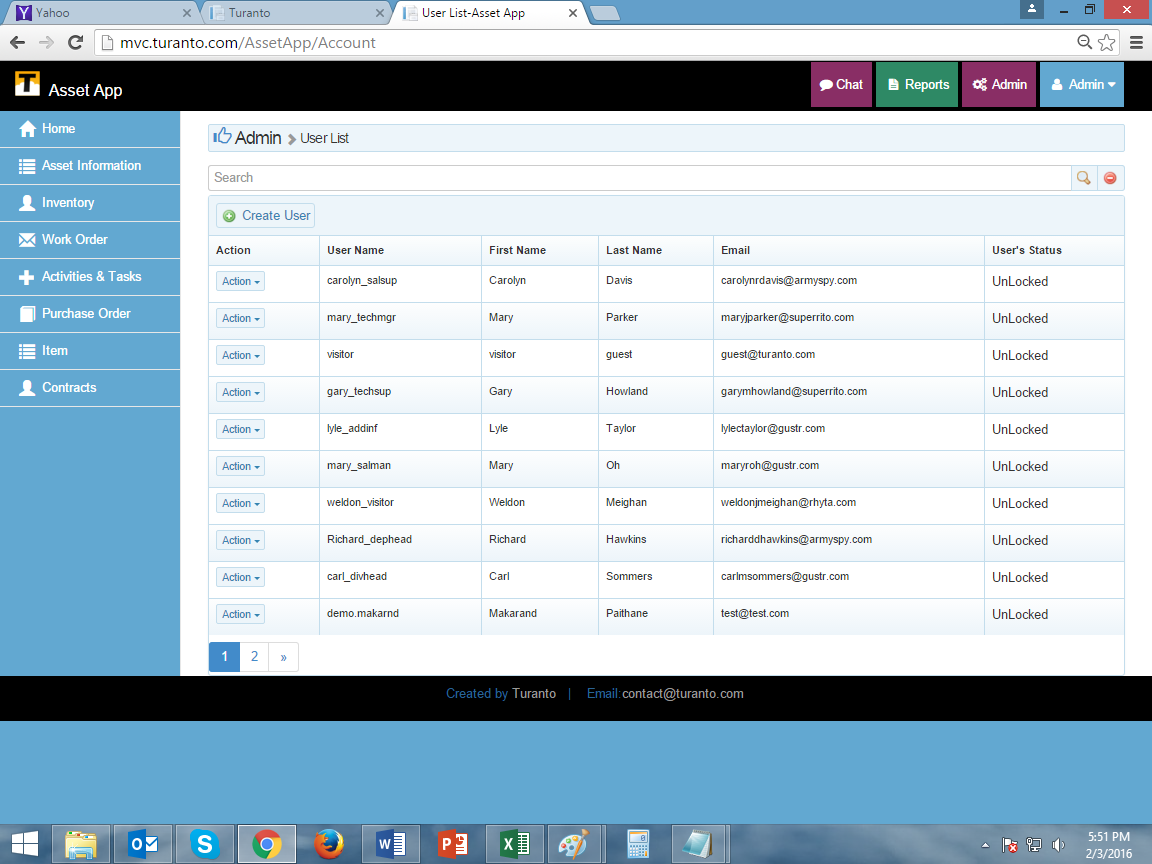


TODO: Replace this Image

Figure 16: Add user (By Admin)

### Create User in Shared User System

App\_Name allows one password for all applications. Through shared user system, a user ID and password can be used to access all the applications. The users for shared user system cannot register themselves and only admin can create these users (Figure 16).

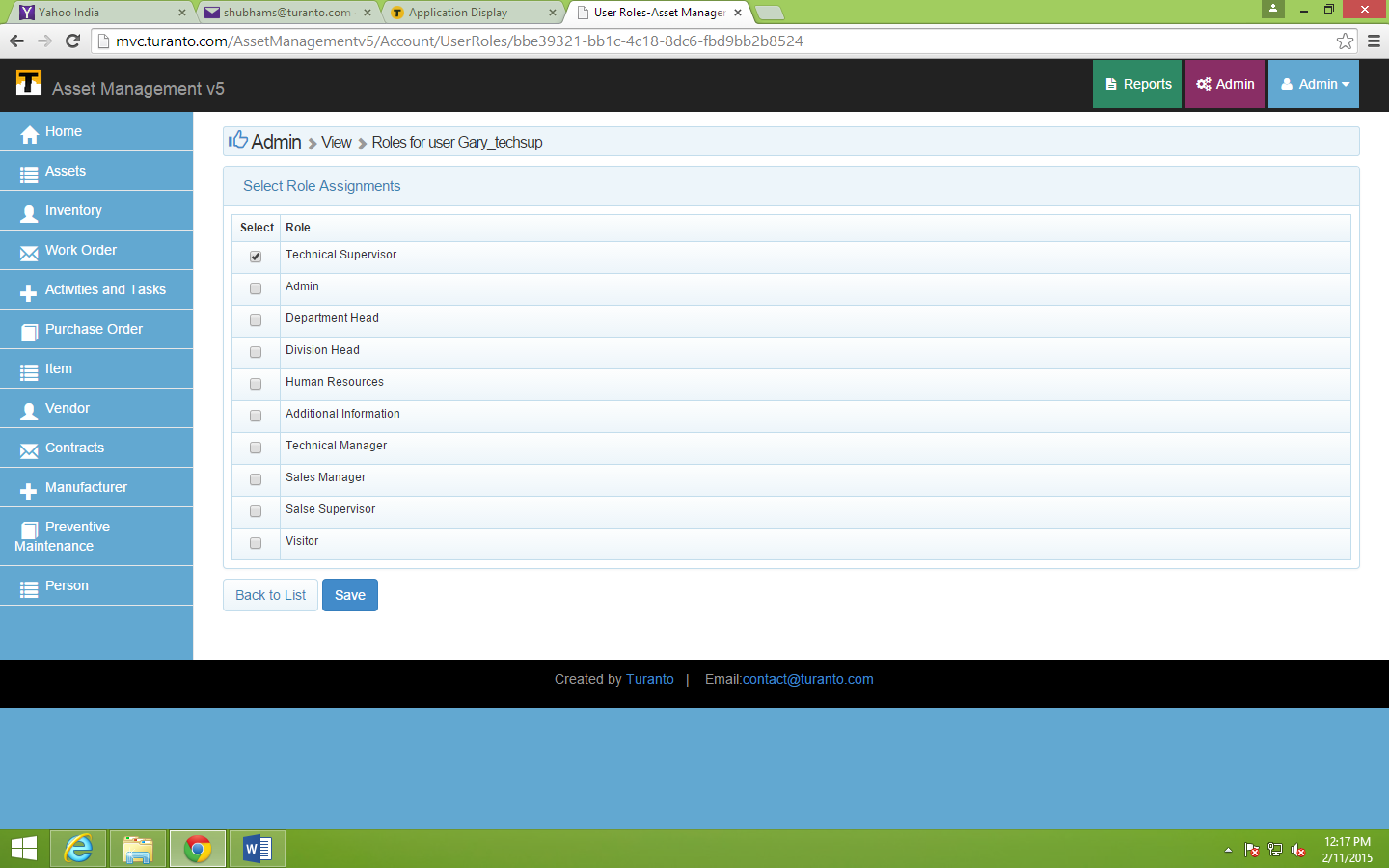


TODO: Replace this Image

Figure 17: User List in ‘Asset App’ Application

### Method 1: Assign Roles to User

The action button in user list (See Figure 17) has 3 options. The first option is to select roles for the user. Select the respective role for your user.



TODO: Replace this Image

Figure 18: Assigning ‘Technical Supervisor’ Role for Username- Gary\_techsup

### Method 2: Assign Users to Role

The action button in user list (See Figure 7) has option to assign users. The second last option is to select roles for the user. Select the respective users for your role.

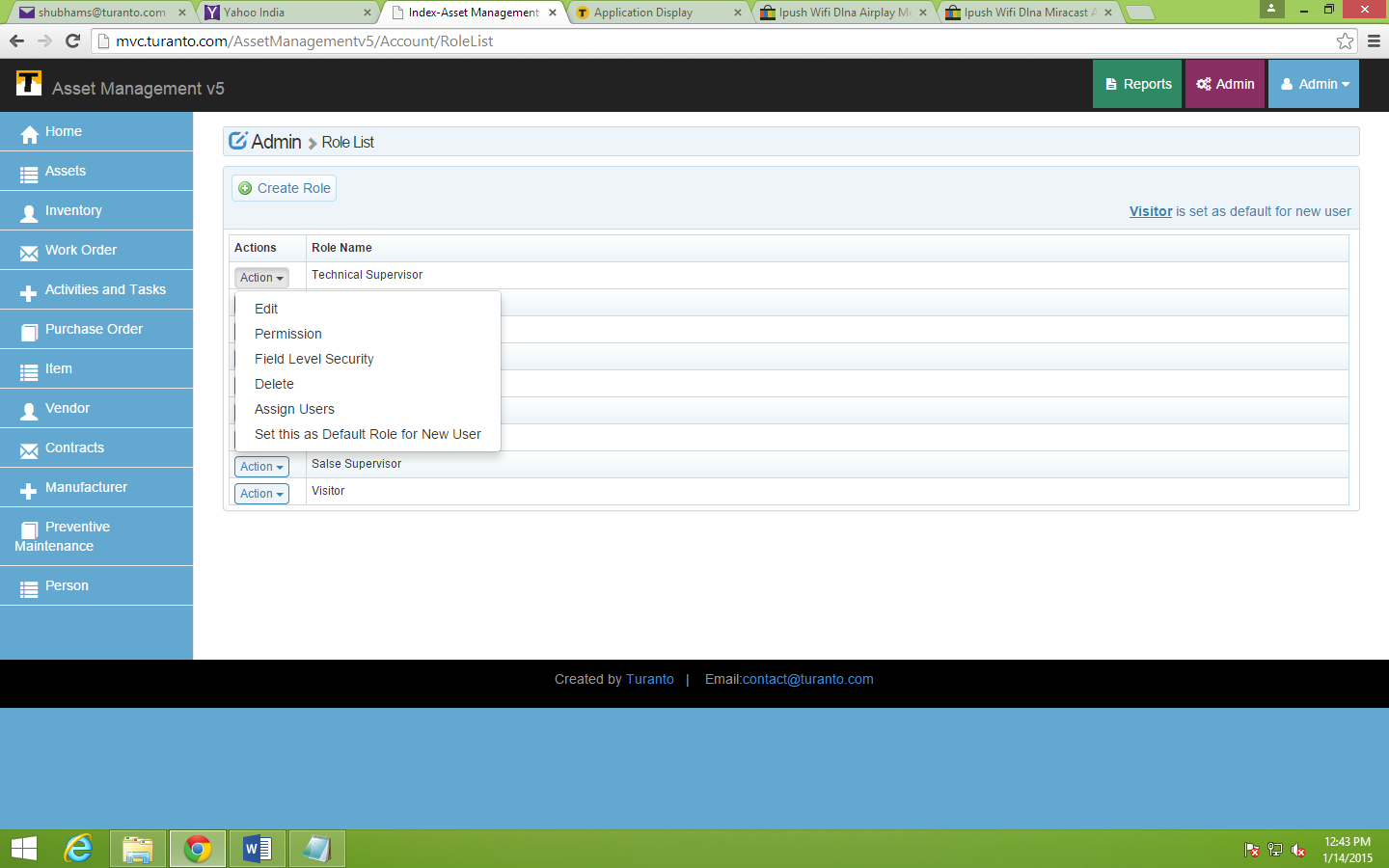
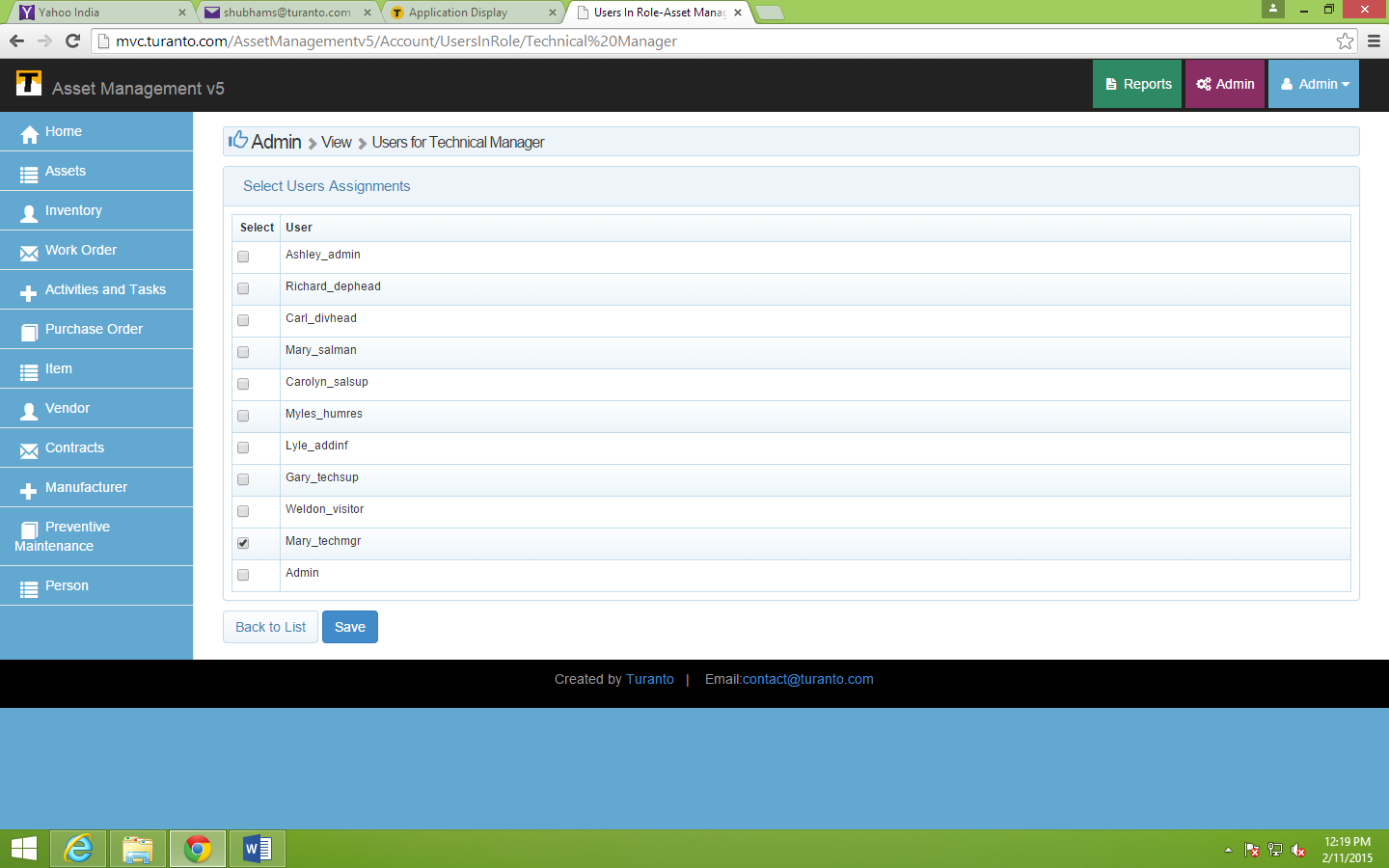


Figure 19: Assign Users to Role from Role List



TODO: Replace this Image

Figure 20: Assigning User ‘Mary\_Techmgr’ to Role- Technical Manager

### Dynamic Roles

Dynamic roles allow assigning a role to a user based on the value of entity column. For example if an entity name (Entities only associated directly with user are available here) Person has entity column ‘Type’ with value ‘Doctor’, he/she can be map to role ‘Doctors’.

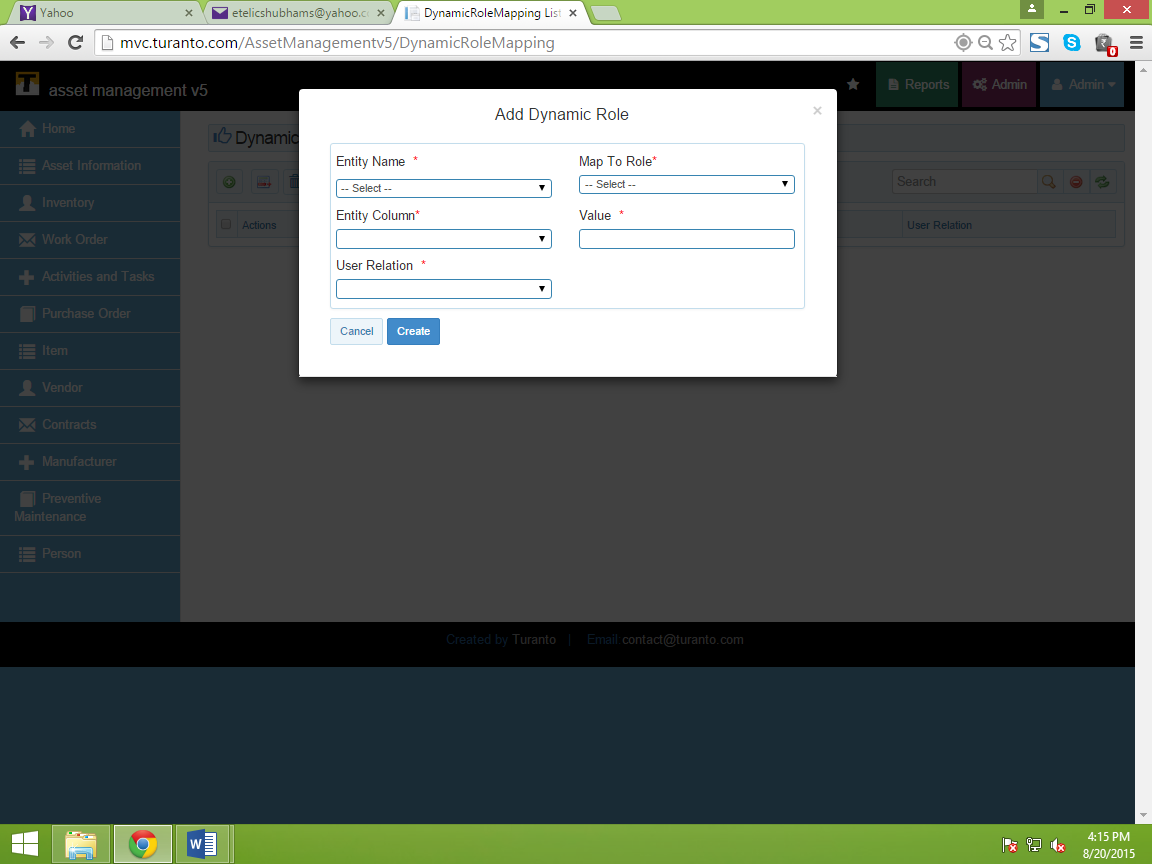


Figure 21: Dynamic Role

Entity Name: Select the name of entity (It should have direct association with user)

Map to Role: Select the role you want to map with this feature

Entity Column: Select the column (property) which should be engaged into this

Value: Select the value which should be present in entity column’s record

User Relation: Select the user login.

# **Test It Out: Using Applications by User Login**

After creating a new role, it is recommended to test it thoroughly before it is assigned to any users. To do this, create a test account and assign the new role to it. Log out as administrator and login as newly created user to test the new role. Alternatively, use a different browser (not a new window in the same browser) to test out the role without logging out as administrator.

The screens below display different security features and role restrictions from the perspective of users. As all the screenshots of all the users is not possible, the figures below are selected to display consequences of Restricting Access by using role based security.

**Note: The new versions of App\_Name may have different screens. The difference exists only in viewing the application without any changes in functional aspects. To see the current screen, check ‘common user’s guide’.**

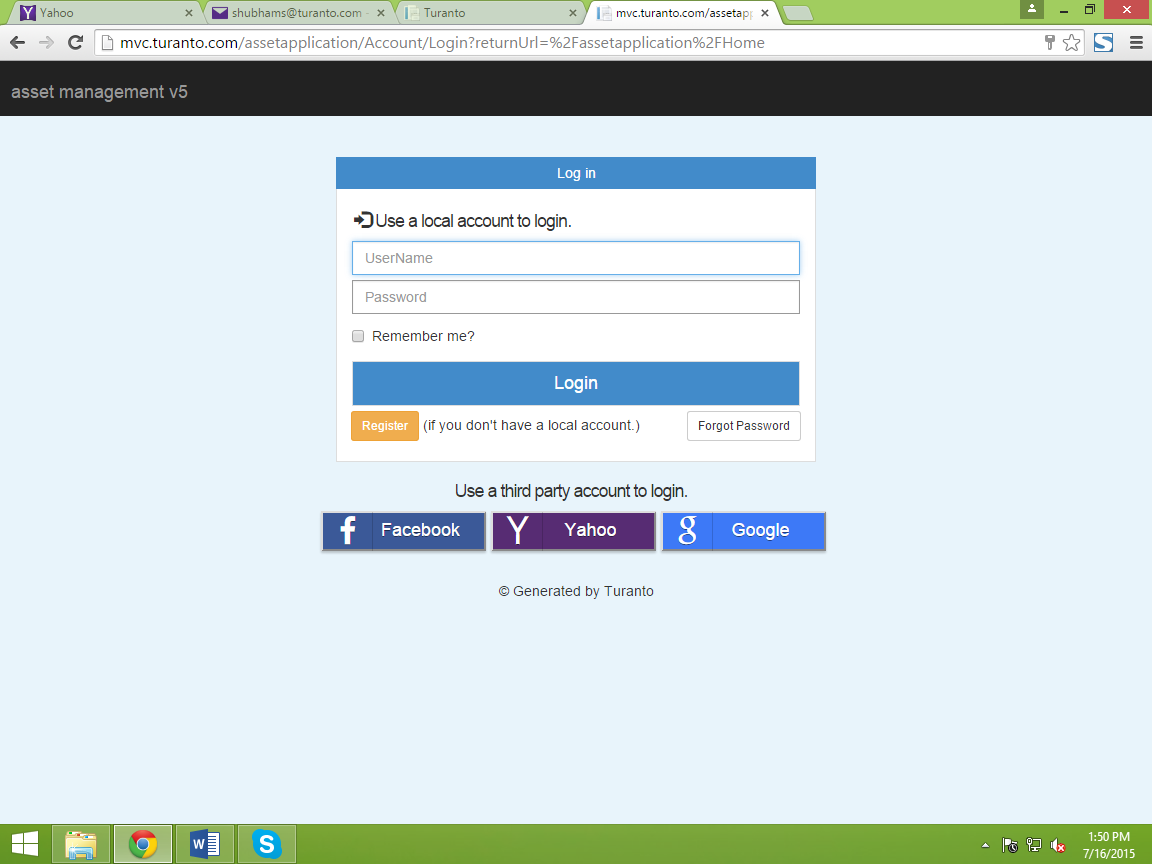
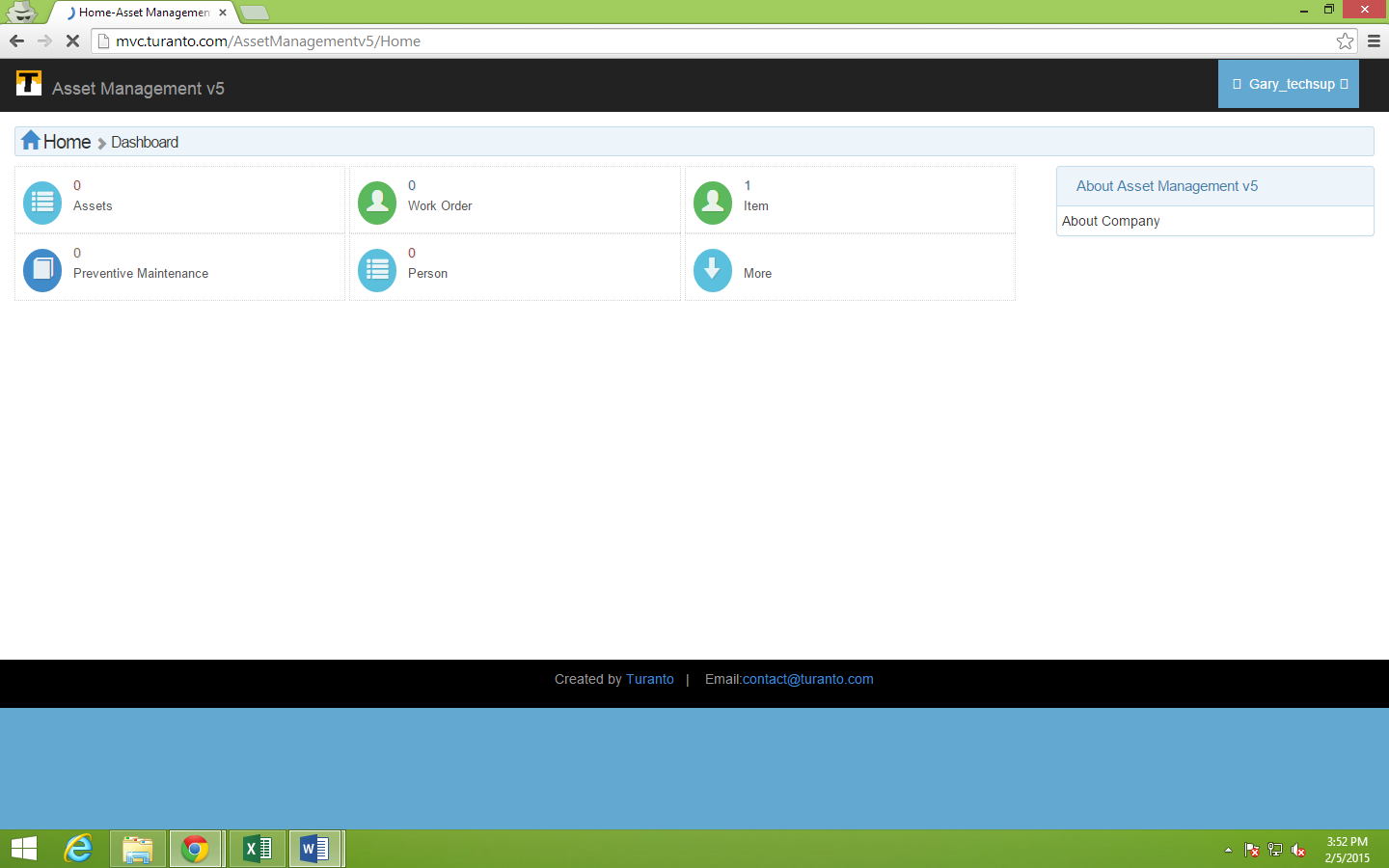


Figure 22: User Login Screen (Common to all Users)

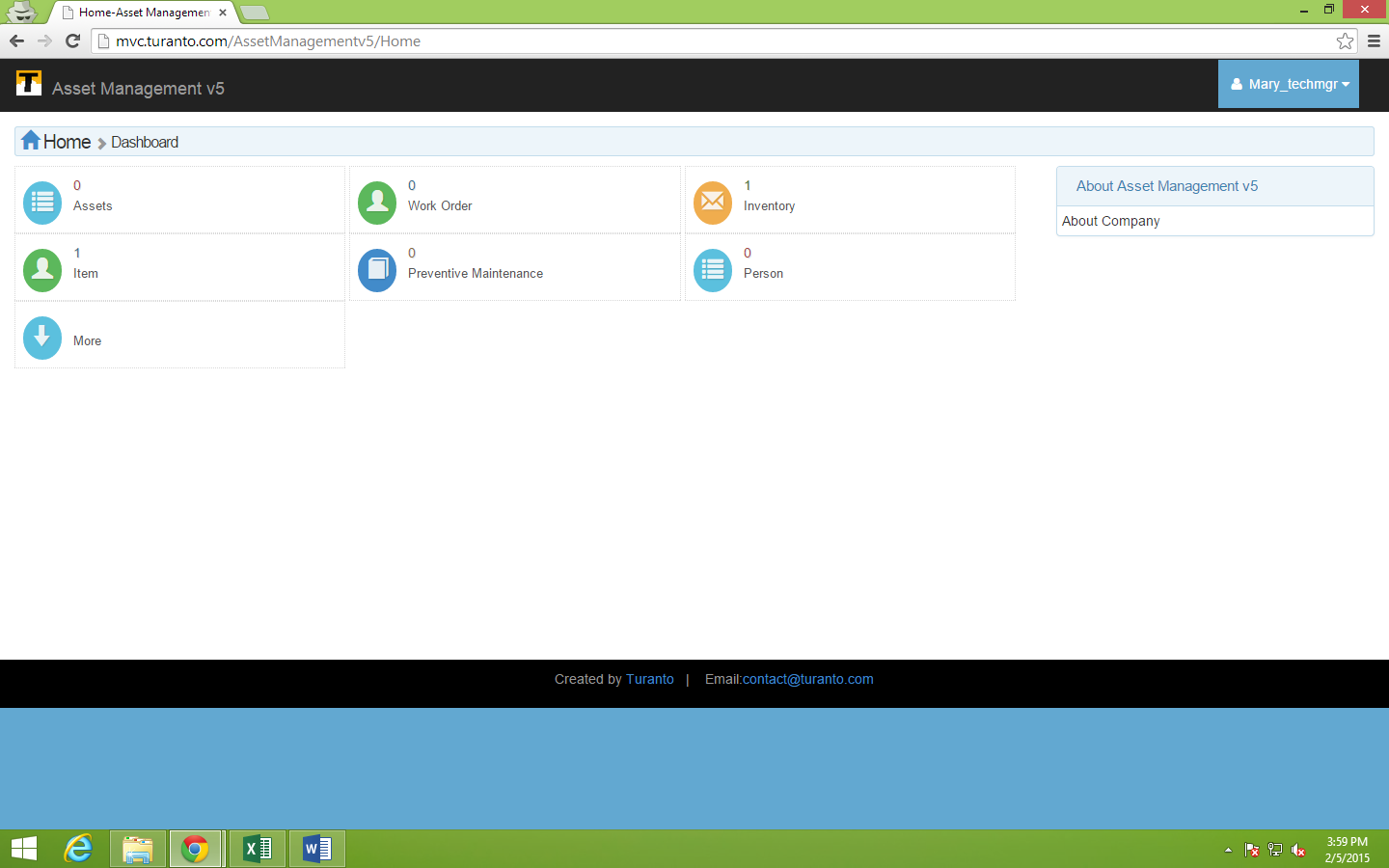
## Home Screen For Users



TODO: Replace this Image

Figure 23: ‘Home’ View of Technical Supervisor

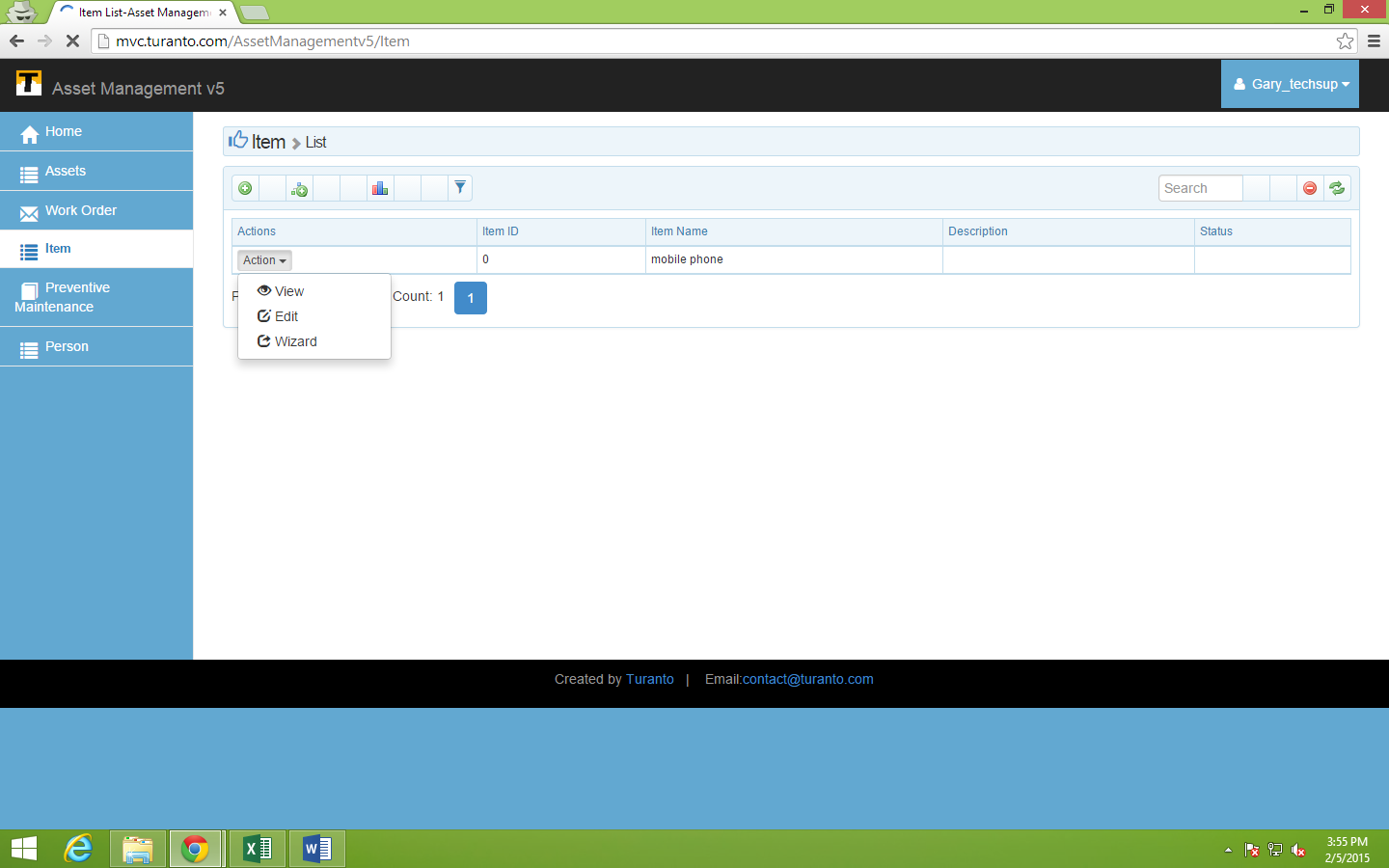
The technical supervisor cannot see the ‘Inventory’ entity! But the technical manager can see.



TODO: Replace this Image

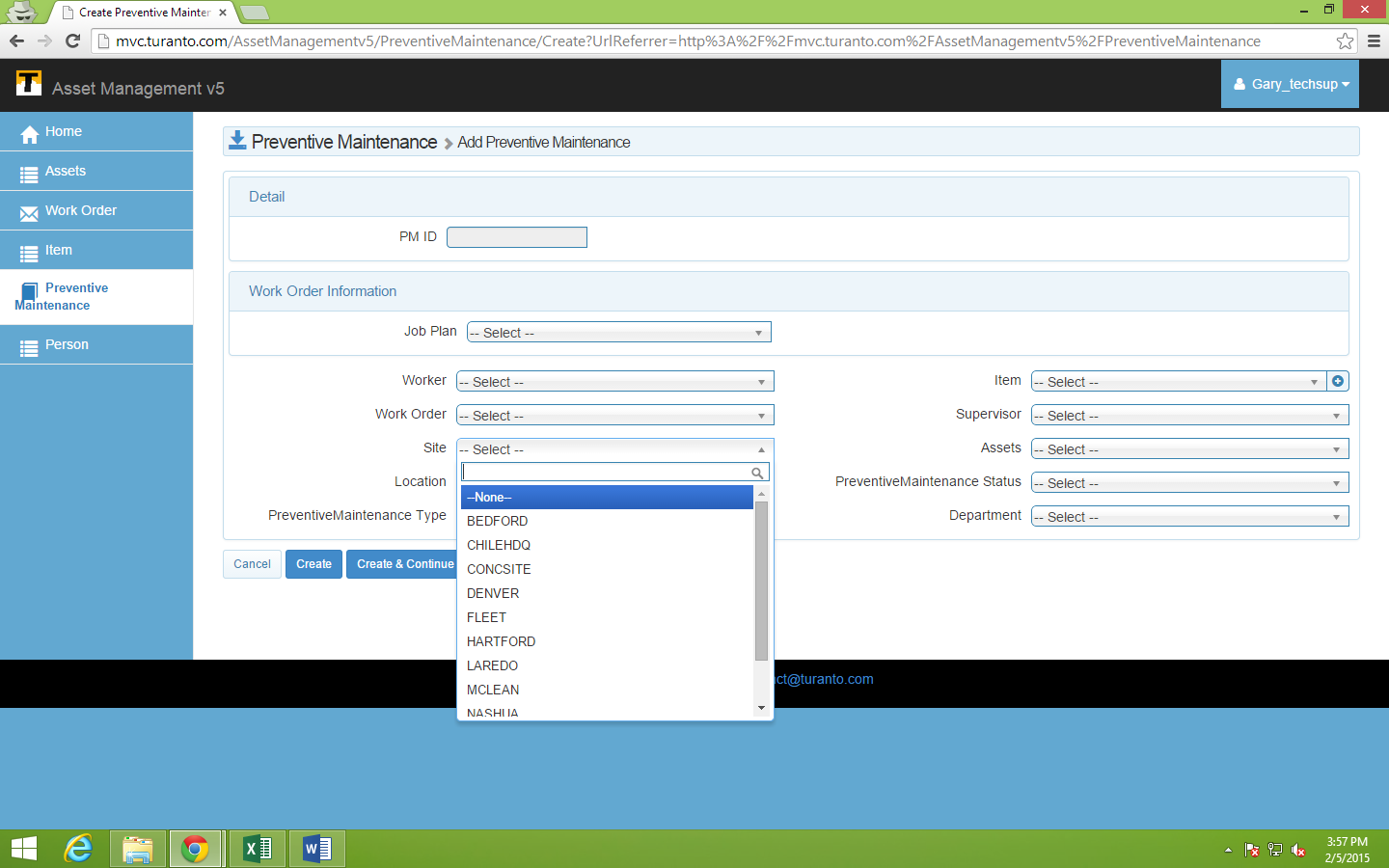
Figure 24: Home View of Technical Manager

## User- Technical Supervisor



TODO: Replace this Image

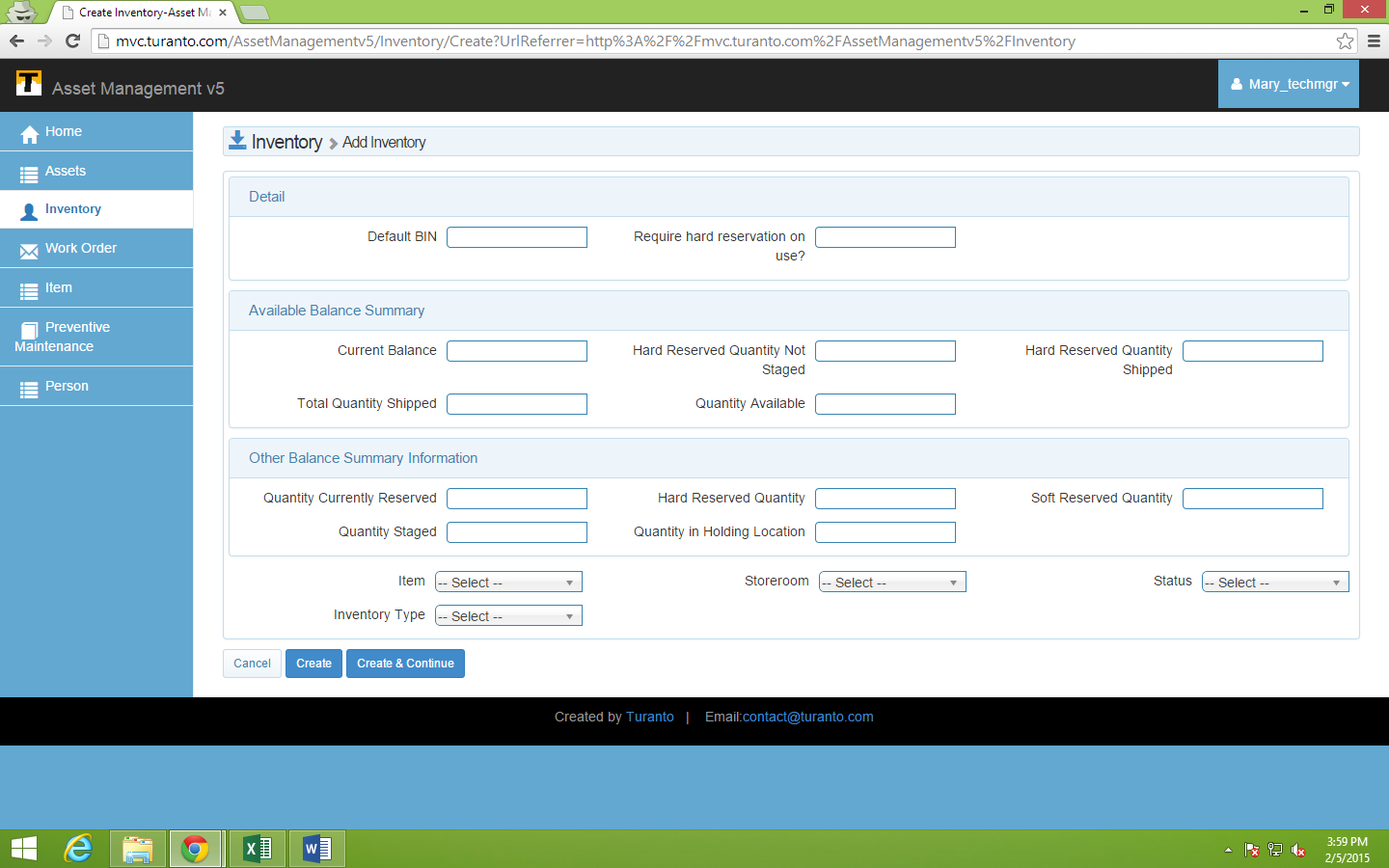
Figure 25: View Permissions were given to ‘Asset’ Entity (Delete option not available!)



TODO: Replace this Image

Figure 26: Adding a New Record in ‘Preventive Maintenance’

## User- Technical Manager



TODO: Replace this Image

Figure 27: Adding Record in ‘Inventory’ Entity

# User Based Security

In multi-tenant environment there is a need for very fine-grained controls for user access to records. With the User-level security feature in App\_Name application you can allow access based on:

* Records that were created by the current user (record owner)
* Records that were assigned to the user
* Records that are associated/related to the user

This is the most restrictive of access levels and especially useful to isolate reports and data for each customer, group or tenant in a multi-tenant environment.

For example, in a sales report that gets shared with regional manager of a particular region say Western, you may want to allow only the western regional manager to see all the western sales data and allow each sales person of that particular region to see only the accounts data he or she owns or that assigned to him/her to manage.

## How User Based Security works?

Following example demonstrates how user based security is applied.

* The entities that have direct association with user are displayed in top section (Section 1 in Figure 28)
* The entities that are associated indirectly with user are displayed prominently. (Section 2 in Figure 28)
* You can select one entity of top section to apply user based security.
* The entities associated with top entity (directly or indirectly) will be secured by default.
* You can lift the security from roles in ‘Ignore for Roles’ column.

TODO: Replace this Image



Top Section

Associated Entities

Figure 28: User Based Security

# Application Configuration Settings

Application configuration settings is an informatory section for developers to attach SSRS Reporting, Active directory settings, file type and file size, and other settings for any other use. Currently, file size is restricted to 5 MB and file types supported are: xlsx, xls, docx, txt, png, gif, jpg, jpeg, bmp, pptx, ppt, pdf.

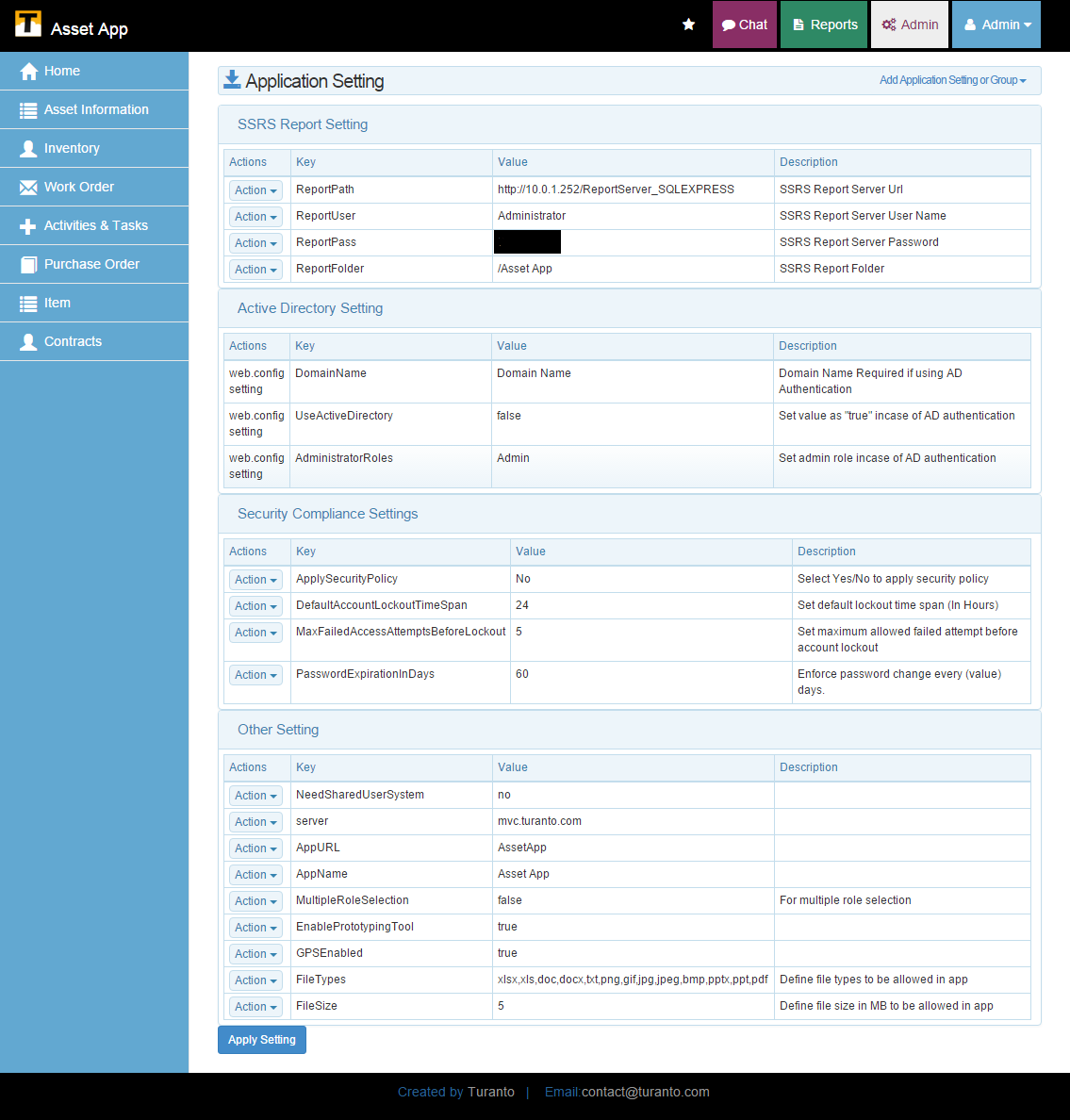


Figure : Application Setting Configuration

SSRS Reporting Settings:

These settings are the credentials to set up SSRS Reporting tool with your application.

## Active Directory Settings

These are the credentials to configure active directory settings with application.

## Security Compliance Settings

Application administrators can control the user’s access to application through application configuration settings. The users can be restricted for further login attempt upon specific times of failed login attempts, locking the user-id for particular period of time and enforce password change over a period of cycle.

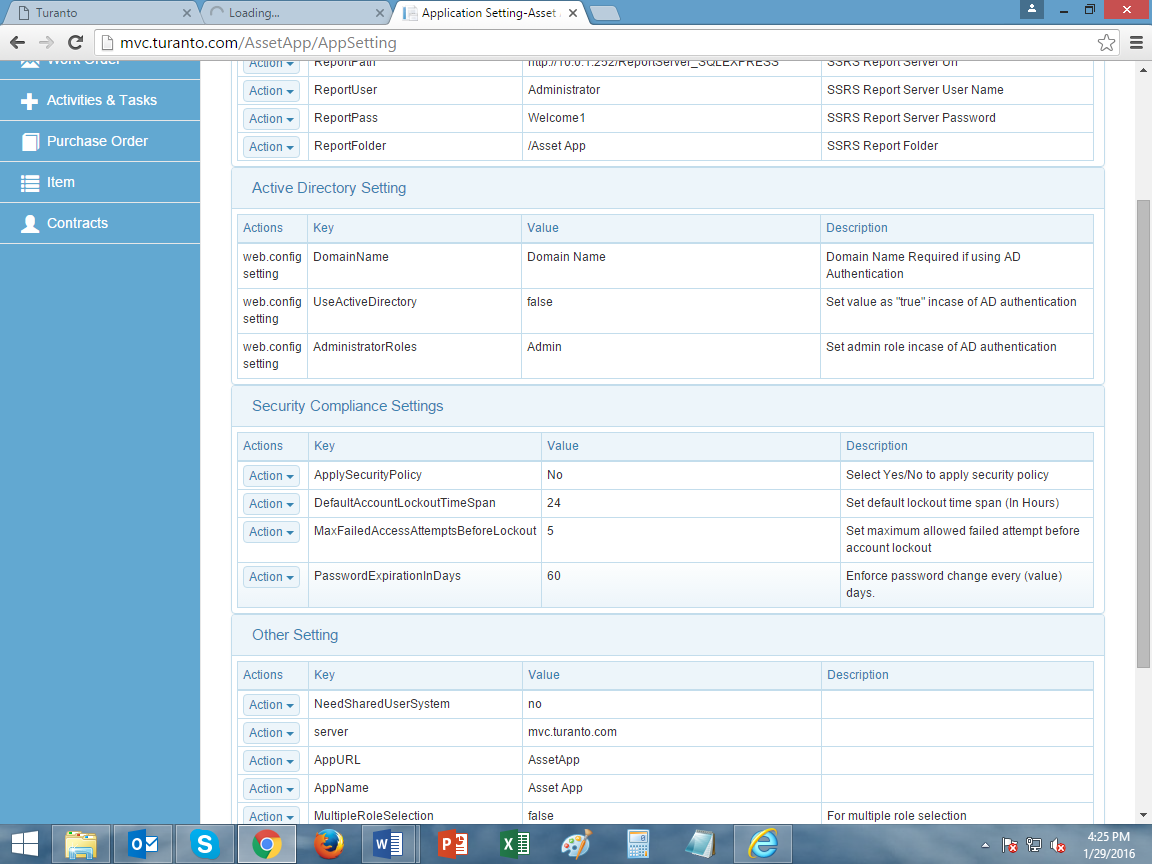


Figure : Security Compliance Settings for User Login

## Other Settings

The settings are for:

* Need Shared User System: With shared user system, users created in one application can access another application of same organization. The user(s) needs to be provided with a role for accessing the content of that application.
* Server: The information here will be reflected as server link in emails generated from application. It has importance if the application is hosted on third party servers, the server name should be replaced here.
* App URL: Similar to above, the URL is used to create link in emails. It has high importance if application is hosted on any server other than Company\_Name’s server.
* App Name: Similar to above, the application name in the mails generated from application can be set from here.
* Multiple Role Selection: This enables/disables multiple role selection feature in application.
* Enable Prototyping Tool: It enables/disables feedback form generated on right click of mouse.
* GPS Enabled: It enables/disables your browser to track your location for application.
* File Types: You can define the supported formats of files over here.
* File Size: You can define the maximum limit of file size supported in application.

## Add Application Settings or Group

You can create your own set of application setting information list other than these default settings. Also you can edit the existing settings to change their values.

Add Application Setting Group: This option allows you to create a new group for application settings configuration.

Add Application Setting: This option allows you to add new application setting through Key (Mandatory), Value (Mandatory) and Description.

# Multi-Tenant Security

To understand multi-tenant security let’s take an example of organization. Sometimes organizations need a more complex way to set up multi-tenant security and assign different rights and permissions to users in the different offices. (See Figure 31).

Example:

* National organizations with a complex hierarchy of headquarter regional offices, branch offices and affiliates.
* A university may have one central university foundation with separate offices representing the different colleges beneath it.

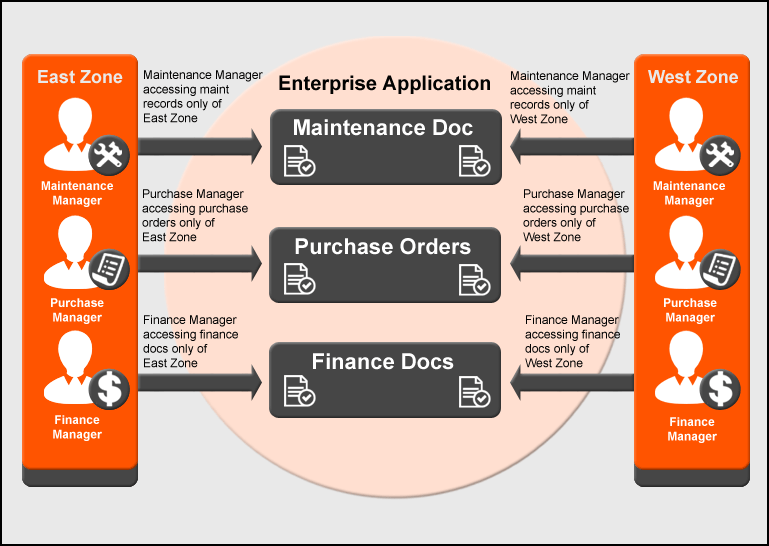


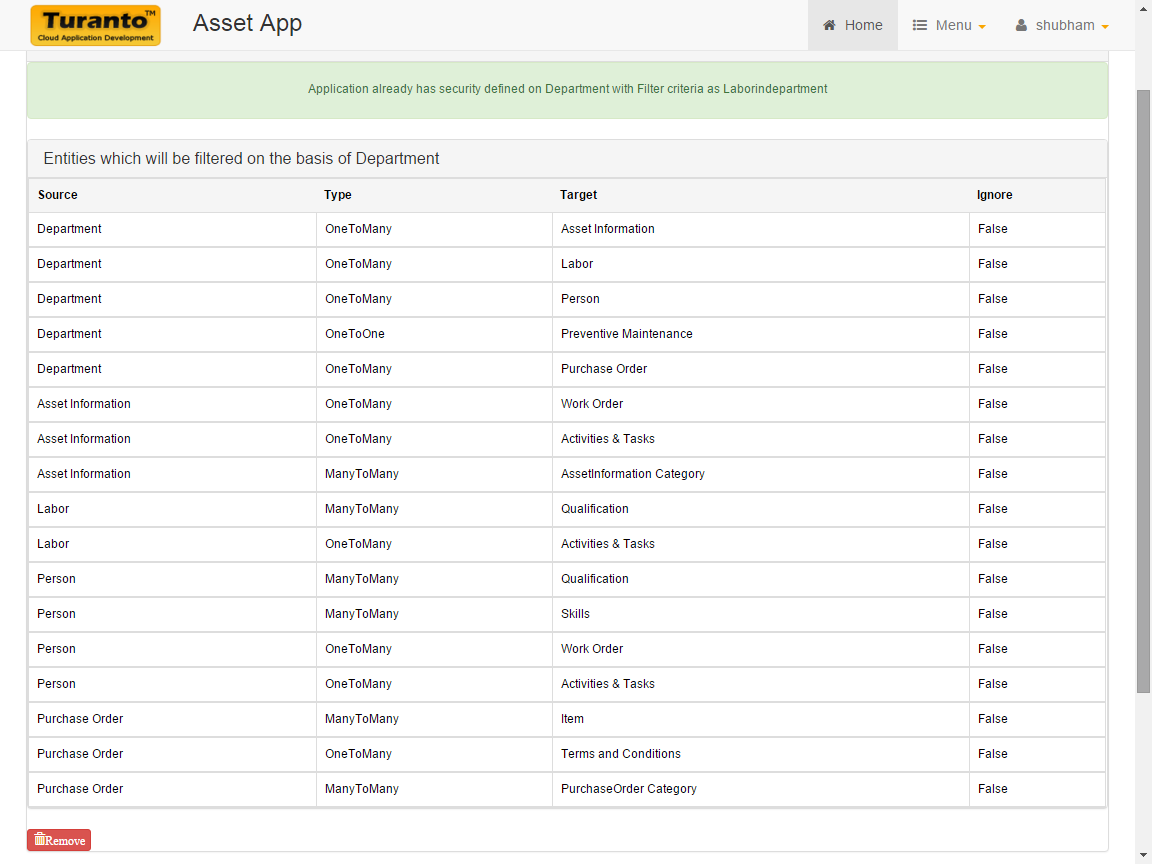
Figure 31: Organization Level Security Model

In above figure, an organization has two branch offices (let’s say: East and West) and a head office. The sales report of East branch office should be accessible only to regional managers of East branch and head office. Similarly, the sales report of West branch should be accessible to regional managers of West branch and head office. Instead of creating a separate view for each manager, organizational level security allows regional managers to see data of only for a particular region. This organizational level security is generalized in App\_Name and termed as multi-tenant security.

In App\_Name application, the multi-tenant security is created using Model Generator. To create multi-tenant security follow the following steps:

## Steps In Model Generator

1. Select the entity on which you want to apply multi-tenant security.
2. Select the filter criteria from dropdown. (The entity/association in filter criteria reflects the entity that is directly associated with user).
3. From the list of entities falling under multi-tenant security, select the entity that you want to ignore.
4. Click Save.



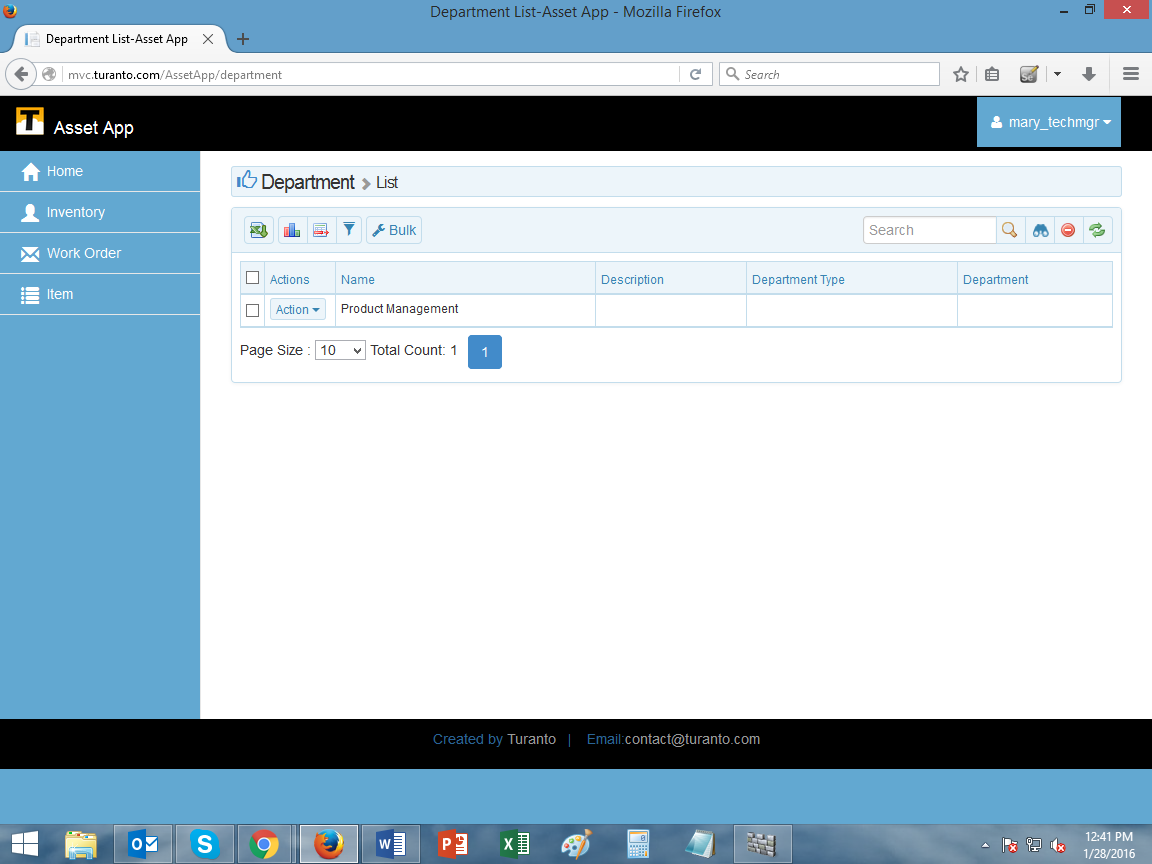
TODO: Replace this Image

Figure 32: Security Setting in App\_Name Application Generator

The Figure 32 shows the associations created in application (used in multi-tenant security), the entity selected for multi-tenant security and the main entity that was directly associated with user. The application is launched and following screen demonstrates effects of security settings.

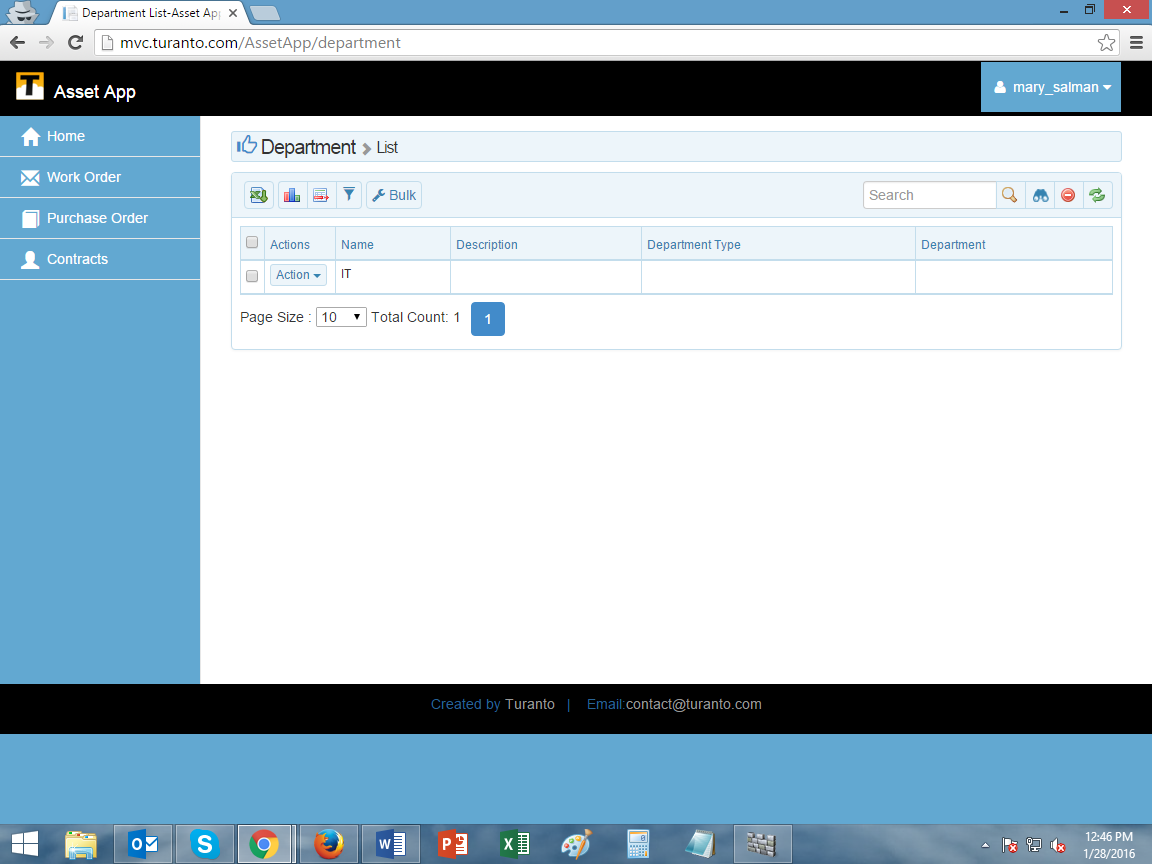
# Test It Out

## Entity Visible



TODO: Replace this Image

Figure 33: The users can see only their department: Mary\_techmgr user can see only her department



TODO: Replace this Image

Figure 34: The users can see only their department: mary\_salman can see only her department

Note: The security options in App\_Name application do not overrule each other. The screens above are of users assigned in a role and only permitted entities are visible (along with options of CRUD, self-service and disabled verbs). If a user has rights to view an entity, then following possibilities would be present:

Table 6: Security Options in Generated Application

|  |  |
| --- | --- |
| User Access Levels | Security Options and Combinations |
| He has rights to add edit delete all records | Normal entity permissions |
| He cannot see particular properties | Field level security |
| He can add edit delete his own records but can only view other people’s records. | Self-service |
| He can CRUD the records of his company only | Multi-Tenant Security is applicable on that entity |
| He can CRUD his own records but see other records of his company | Multi-Tenant security + Self-service |
| He can CRUD his own records | User based security |
| He can CRUD records of people below him | Hierarchy enabled in user based security |

The above table is simple example of various security combinations. End application can have different combinations also.

# USER DEFINED PAGES

Do you want a different home page for every role? App\_Name provides option to create a ‘User Defined Pages’ to differentiate the home screen for every user role. This feature of App\_Name has following benefits:

1. It saves ample amount of time required to customize home page of application.
2. Every application role will see only his home page.
3. The admin can display relevant information to be conveyed (for example: about application, user role, functions, responsibilities etc.)
4. The home page is available in rich text format.
5. The Rich text box contains options to insert photos, videos and links to another locations.
6. The options in rich text box allows to format data in a professional manner.

## Creating User Defined Page in App\_Name

### Select Page

Select a page you want to edit. ‘Quick Add’ menu provides option to create a new page by giving a name.

### Select Roles

Displays all the roles of application. Select the roles to whom you want to display the page.

### Custom Information

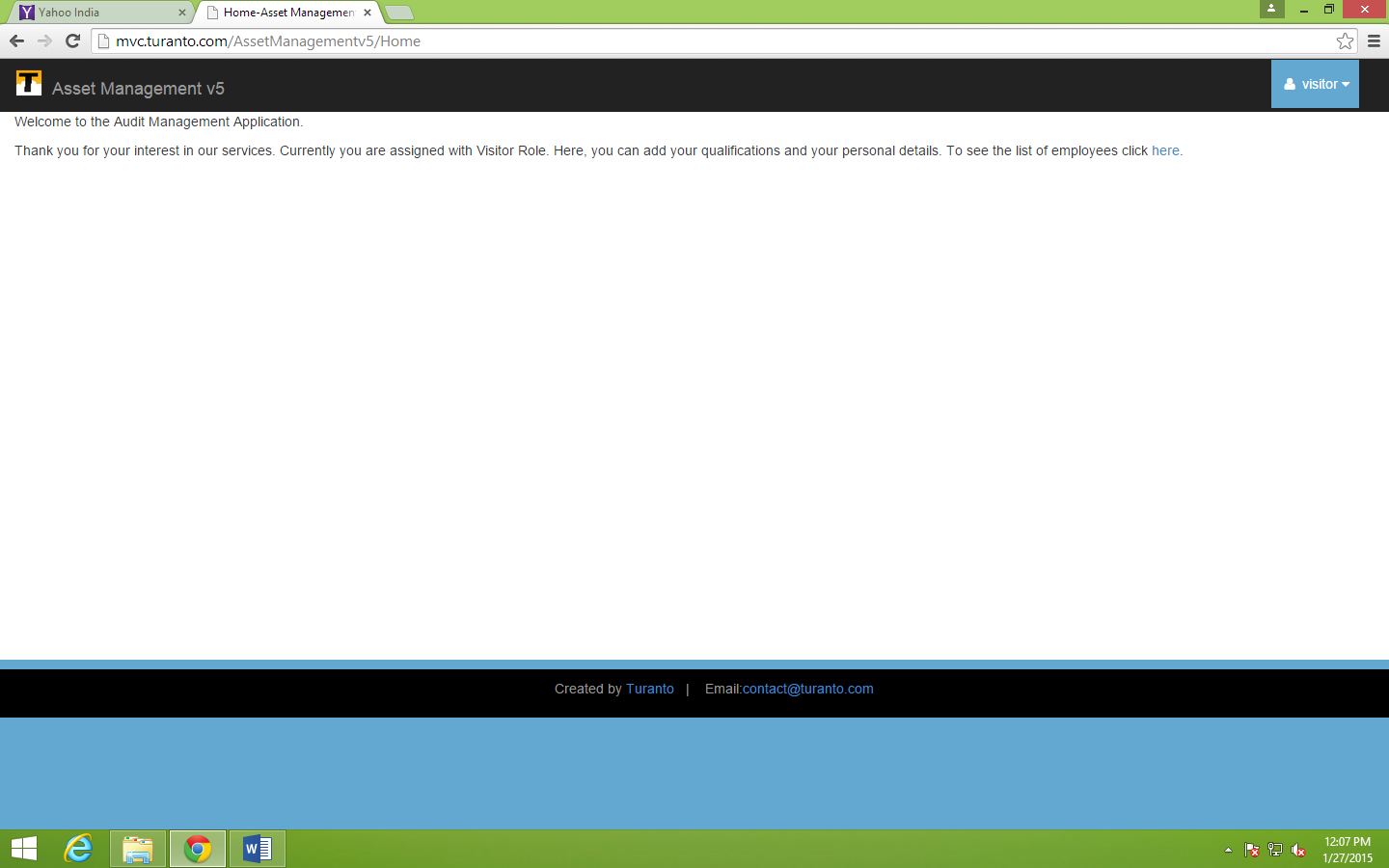
This is the information that will be displayed as body. The option has WYSIEYG editor for rich text formatting.



TODO: Replace this Image

Figure 35: User Defined Page

### TEST IT OUT



TODO: Replace this Image

Figure 36: Home Page for Visitor Login