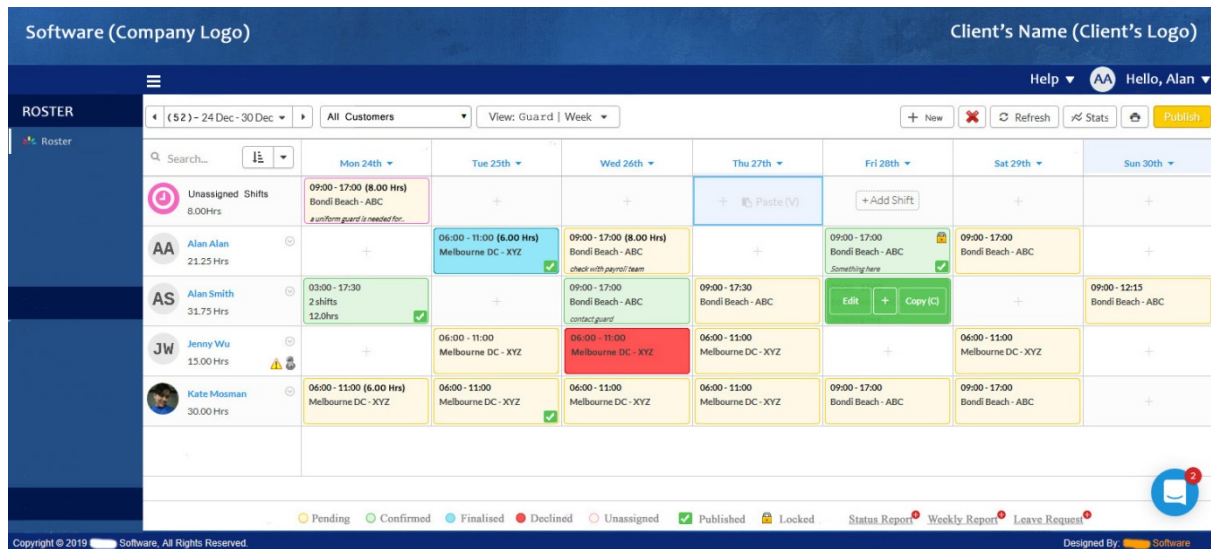


# Roster Application Layout

The following screen shows the required page layout



## 1. Login Process

### 1.1 Web Application Login Process

A member enters a username and password (username: [john.smith@OSG](mailto:john.smith@OSG))

Extract the alias name after the @ symbol (OSG)

Query to check the client status `SELECT clntID,clDB,sbpHide,sbpExp FROM sgClients WHERE clAlias = ['alias_name']`

If not found, display the following message (*Invalid username or password!*).

If `sbpHide` = true then display (*Inactive subscription, contact your system administrator!*)

Check the subscription expiry date, if `sbpExp` is expired then display (*Inactive subscription, contact your system administrator!*)

If all ok then store the `clntID` and `clDB` in a global session variables and connect to client database

Use the following query to check if the username and password are correct

`SELECT * FROM usrMgr WHERE clntID = $_SESSION['clntID'] AND usrID = ['user_name'] AND usrPass = ['md5_password']`

If not found, display (*Invalid username or password!*) If all ok then proceed to the main page based on the user's access rights

## 2. Roster Calendar Display

### 2.1 Preparing and Loading Roster

Pre-filling roster week drop down list, Use the following query:

```
SELECT TOP 20 rsID,rsWkNo,rsDFrom,rsDTo FROM wsRoster WHERE cIntID =  
$_SESSION['cIntID'] ORDER BY rsYear DESC, rsWkNo DESC
```

**cIntID** should be read from the global session variable which is stored when the user is logged in  
**rsID** value should be stored to match the selected roster week drop down list value and should be hidden to the user

Detect the current week based on server's date and then select the current roster week

If no roster week found (new client) then the list should be empty and user must use the New button to create new roster week (Refer to creating New Roster Week section)

### 2.2 Displaying the selected roster week Shifts

Extract the roster id **rsID** from the drop down list selected item

Pre-fill the customer list for search purpose using the following query:

```
SELECT DISTINCT (wsRstDet.cusID),Customers.cusID,Customers.cusName FROM  
(wsRstDet INNER JOIN Customers ON wsRstDet.cusID = Customers.cusID AND  
wsRstDet.cIntID = Customers.cIntID) WHERE cIntID = $_SESSION['cIntID'] AND rsID  
= $_POST['rsID']
```

Make sure to add the item "All Customers" to customers drop down list and set it as the default

Use the following query to pull the roster week start and end dates

```
SELECT rsDFrom,rsDTo FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND rsID  
= $_POST['rsID']
```

Use **rsDFrom** date value to fill the first calendar day and then increase by 1 to fill the remaining 6 days

There are two types of views available to the user (by Guards and by Location)

#### 2.2.1 View Shifts by Guards

	Mon 24th	Tue 25th	Wed 26th	Thu 27th	Fri 28th	Sat 29th	Sun 30th
Unassigned Shifts 8.00Hrs	09:00 - 17:00 (8.00 Hrs) Bondi Beach - ABC <i>a uniform guard is needed for...</i>	+	+	+	+ Paste (V)	+ Add Shift	+
AA Alan Alan 21.25 Hrs	+	06:00 - 11:00 (6.00 Hrs) Melbourne DC - XYZ	09:00 - 17:00 (8.00 Hrs) Bondi Beach - ABC <i>check with payroll team</i>	+	09:00 - 17:00 Bondi Beach - ABC <i>Something here</i>	09:00 - 17:00 Bondi Beach - ABC	+
AS Alan Smith 31.75 Hrs	03:00 - 17:30 2 shifts 12.0hrs	+	09:00 - 17:00 Bondi Beach - ABC <i>contact guard</i>	09:00 - 17:30 Bondi Beach - ABC	Edit + Copy (C)	+	09:00 - 12:15 Bondi Beach - ABC
JW Jenny Wu 15.00 Hrs	+	06:00 - 11:00 Melbourne DC - XYZ	06:00 - 11:00 Melbourne DC - XYZ	06:00 - 11:00 Melbourne DC - XYZ	+	06:00 - 11:00 Melbourne DC - XYZ	+
Kate Mosman 30.00 Hrs	06:00 - 11:00 (6.00 Hrs) Melbourne DC - XYZ	06:00 - 11:00 Melbourne DC - XYZ	06:00 - 11:00 Melbourne DC - XYZ	06:00 - 11:00 Melbourne DC - XYZ	09:00 - 17:00 Bondi Beach - ABC	09:00 - 17:00 Bondi Beach - ABC	+

First row must contain the unassigned shifts by using the following:

```
SELECT wsRstDet.rsID,wsRstDet.rsSTime,wsRstDet.rsETime,wsRstDet.rsHours,
wsRstDet.rsNotes,wsRstDet.rsStatus, wsLoc.sName,wsLoc.sCode FROM (wsRstDet
INNER JOIN wsLoc ON wsRstDet.sID = wsLoc.sID AND wsRstDet.cIntID =
wsLoc.cIntID) WHERE wsRstDet.cIntID = $_SESSION['cIntID'] AND wsRstDet.rsID =
$_POST['rsID'] AND wsRstDet.rsNoGrd = True AND wsRstDet.rsDate = # & ['wkDay']
# ORDER BY wsRstDet.rsSTime ASC
```

If “All Customers” is selected then use the following query:

```
SELECT DISTINCT (wsRstDet.grdID),Guards.gFName,Guards.gLName,Guards.gMob
FROM (wsRstDet INNER JOIN Guards ON wsRstDet.grdID = Guards.grdID AND
wsRstDet.cIntID = Guards.cIntID) WHERE wsRstDet.cIntID = $_SESSION['cIntID']
AND wsRstDet.rsID = $_POST['rsID'] ORDER BY Guards.gFName,Guards.gLName
```

Else if an individual customer is selected then extracts the customer ID from the drop down list selected item and use the following query:

```
SELECT DISTINCT (wsRstDet.grdID),Guards.gFName,Guards.gLName,Guards.gMob
FROM (wsRstDet INNER JOIN Guards ON wsRstDet.grdID = Guards.grdID AND
wsRstDet.cIntID = Guards.cIntID) WHERE wsRstDet.cIntID = $_SESSION['cIntID']
AND wsRstDet.rsID = $_POST['rsID'] AND wsRstDet.cusID = ['cusID'] ORDER BY
Guards.gFName,Guards.gLName
```

Note: For guards view, you must check if the security licence is expired or not found.

```
SELECT licExp FROM grdLics WHERE licID = 1 AND cIntID = $_SESSION['cIntID'] AND
grdID = ['grdID']
```

Add a warning symbol to the guard section if expired or not found

Once you have a list of all guards, you need to pull all shifts by looping through each day for the current selected roster week or as your preferred method

Reading guards shifts and displaying the details on the roster calendar

```
SELECT wsRstDet.rsID,wsRstDet.rsSTime,wsRstDet.rsETime,wsRstDet.rsHours,
wsRstDet.rsNotes, wsRstDet.wsStatus,wsRstDet.rsLock,wsRstDet.rsPub,
wsLoc.sName,wsLoc.sCode FROM (wsRstDet INNER JOIN wsLoc ON wsRstDet.sID =
wsLoc.sID AND wsRstDet.cIntID = wsLoc.cIntID) WHERE wsRstDet.cIntID =
$_SESSION['cIntID'] AND wsRstDet.rsID = $_POST['rsID'] AND wsRstDet.grdID =
['grdID'] AND wsRstDet.rsDate = # & ['wkDay'] # ORDER BY wsRstDet.rsSTime ASC
```

## 2.2.1 View Shifts by Locations

The screenshot displays a web-based roster management application. At the top, there's a navigation bar with a user profile 'Hello, Alan' and various icons. Below this is a header section with filters: '(52) - 24 Dec - 30 Dec', 'All Customers', and 'View: Location | Week'. A toolbar contains buttons for '+ New', a delete icon, 'Refresh', 'Stats', and 'Publish'. The main area is a calendar grid for the week of December 24th to 30th. It shows shifts for two locations: 'Bondi Beach - ABC' and 'Melbourne DC - XYZ'. Shifts are assigned to guards: Alan Alan, Alan Smith, Jenny Wu, and Kate Mosman. Each shift entry includes a time slot (e.g., 09:00 - 17:00) and a duration (e.g., 8.00 Hrs). A sidebar on the left lists the guards with their total hours. At the bottom, there's a status legend with icons for Pending, Confirmed, Finalised, Declined, Unassigned, Published, and Locked. Navigation links for 'Status Report', 'Weekly Report', and 'Leave Request' are also present.

If “All Customers” is selected then use the following query:

Else if an individual customer is selected then extracts the customer ID from the drop down list selected item and use the following query:

Once you have a list of all locations, you need to pull all shifts by looping through each day for the current selected roster week or as your preferred method

```
SELECT wsRstDet.rsDID,wsRstDet.rsSTime,wsRstDet.rsETime,wsRstDet.rsHours,
,wsRstDet.rsNotes,wsRstDet.wsStatus,wsRstDet.rsLock,wsRstDet.rsPub,Guards.gFNam
e,Guards.gLName FROM (wsRstDet INNER JOIN Guards ON wsRstDet.grdID =
Guards.grdID AND wsRstDet.clnID = Guards.clnID) WHERE wsRstDet.clnID =
$_SESSION['clnID'] AND wsRstDet.rsID = $_POST['rsID'] AND wsRstDet.sID =
['sID'] AND wsRstDet.rsDate = # & ['wkDay'] # ORDER BY wsRstDet.rsSTime ASC
```

Here are few points on how to display and process the shifts

- ## Cell Cursor Actions

Blank cell:

If there is a copied shift then display the following:

Paste (V):                      Copy the current shift

Edit: Edit the current shift

+: Add new shift

Copy(C): Copy the current shift

## 2.3 Create New Roster Week

The first screenshot shows the 'New Roster Week' dialog with 'Start Date' set to 31 Dec 2018 and 'Week no.' 53. It offers two options: 'Blank roster week' or 'Copy from previous week'. The 'Copy from previous week' option is selected, showing 'Roster week' (52) - 24 Dec - 30 Dec and 'Copy to the next' 5 week(s). A 'Continue' button is at the bottom.

The second screenshot shows the 'New Roster Week' dialog with a progress bar indicating 'Date' and 'Location' are completed, and 'Done!' is the next step. It includes a checkbox for 'Include pre-scheduled shifts when available' and a 'Select: All None Inverse' dropdown. Two locations are selected: 'Bondi Beach - ABC' and 'Melbourne DC - XYZ'. 'Cancel' and 'Save' buttons are at the bottom.

The third screenshot shows the 'New Roster Week' dialog with the progress bar at 'Done!'. A 'Shift copy status' section displays a message: 'All shifts have been successfully copied.' A 'Done!' button is at the bottom.

The fourth screenshot shows the 'New Roster Week' dialog with the progress bar at 'Done!'. A 'Shift copy status' section displays a message: 'Copying'. A 'Copying' button is at the bottom.

Follow the next steps to create new roster week

### 2.3.1 Determine the new week start date and week number

Read previous roster weeks to determine the next roster week by using the following query:

```
SELECT TOP 1 rsWkNo,rsDTo FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND  
rsYear = ['Today.year'] ORDER BY rsWkNo DESC
```

If found then fill the start day by increasing the last date of the previous roster week by 1

Else if not found then read previous week from last year by running:

```
SELECT TOP 1 rsWkNo,rsDTo FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND  
rsYear = ['Last.year'] ORDER BY rsWkNo DESC
```

If found then fill the start day by increasing the last date of the previous roster week by 1

If not found then fill the start date with today's date

Calculate the week number based on the start date

### 2.3.2 Populate previous roster weeks

Use the following query to add the previous roster weeks to the copy from drop down list:

```
SELECT TOP 20 rsID,rsWkNo,rsDFrom,rsDTo FROM wsRoster WHERE cIntID =  
$_SESSION['cIntID'] ORDER BY rsYear DESC, rsWkNo DESC
```

User has two options:

- blank roster week (create the new roster week with empty shifts and exit)
- copy from a previous roster week (continue with creating a roster week)

When user clicks on Continue, it brings the location selection page

### 2.3.3 Populate all available locations

Based on the selected previous roster week, use the following query to list all available active locations:

```
SELECT DISTINCT (wsRstDet.sID),wsLoc.sName,wsLoc.sCode FROM (wsRstDet INNER
JOIN wsLoc ON wsRstDet.sID = wsLoc.sID AND wsRstDet.cIntID = wsLoc.cIntID)
WHERE wsRstDet.cIntID = $_SESSION['cIntID'] AND wsRstDet.rsID = $_POST['rsID']
AND wsLoc.isHide = False ORDER BY wsLoc.sName
```

A user must select one or more locations

Continue to next page

### 2.3.4 Copying previous roster week shifts

Get the next **rsID** long integer number by running the following query:

```
SELECT TOP 1 rsID FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] ORDER BY
rsID DESC
```

Increase the roster id **rsID** by 1

Create the new roster week using the new **rsID** value and insert into **wsRoster**

```
INSERT INTO wsRoster ....
```

Set the shift ID **rsDID** to 1 and increase by 1 everytime a shift is successfully inserted

Loop through the number of required next weeks as entered in Page 1

Second loop through each selected location

Use the following query to read all location's shifts from the selected previous roster week

```
SELECT * FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND rsID =
['Copy_rsID'] AND sID = ['Selected_locID'] ORDER BY rsDID
```

Check if the current guard is on leave or not by using the following query:

```
SELECT grdID FROM Guards WHERE cIntID = $_SESSION['cIntID'] AND isHide = False
AND grdID = ['grdID'] AND Not EXISTS (Select lvID FROM grdLve WHERE
Guards.grdID = grdLve.grdID AND (#['rsDate']# BETWEEN grdLve.lvFrom AND
grdLve.lvTo))
```

If the current guard is on leave then set the shift to unassigned by removing all guards related fields

Change the **rsStatus** to Pending on all inserted shifts

Insert this shift into **wsRstDet**

```
INSERT INTO wsRstDet ....
```

Increase **rsDID** by 1 for next shift

### Checking pre-scheduled shifts before moving to next week

Only applies when the user is selected to include the pre-scheduled shifts option

Use the following query to check if there are any pre-scheduled shifts:

```
SELECT * FROM wsRstFut WHERE cIntID = $_SESSION['cIntID'] AND rsDate BETWEEN #
['Roster_StartDate'] # AND # ['Roster_EndDate'] # ORDER BY rsDID
```

If found then check if there is a conflict refer to conflict section

If conflict found then do not add this shift and continue

Check if the guard on leave, refer to previous section

If guard is on leave then add this shift as unassigned

Once all shifts are copied to the first week, increase **rsID** by 1 for second next week if required

Once all required weeks are successfully copied then display the final page

## 2.4 Delete Roster Week

Check if the current logged in user has administrator's rights

If not then display the warning message (Deleting an entire roster requires an administrator privilege. Ask an administrator to delete the current roster week!) and stop the delete process

Check if the roster week is locked by running the following query

```
SELECT rsRecon FROM wsRoster WHERE clntID = $_SESSION['clntID'] AND rsID =  
$_POST['rsID']
```

If rsRecon = True then show warning message (The selected roster week has been finalised and reconciled. Ask an administrator to reset the current roster week!) and stop the delete process

If not locked then display the following confirmation message

(Are you sure you want to delete the selected roster week?)

If Yes then run the following queries to delete the selected shift

```
DELETE FROM wsRstDet WHERE clntID = $_SESSION['clntID'] AND rsID =  
$_POST['rsID']  
DELETE FROM wsRoster WHERE clntID = $_SESSION['clntID'] AND rsID =  
$_POST['rsID']  
DELETE FROM wsRstReps WHERE clntID = $_SESSION['clntID'] AND rsID =  
$_POST['rsID']  
DELETE FROM wsRstGrds WHERE clntID = $_SESSION['clntID'] AND rsID =  
$_POST['rsID']
```

Refresh the entire roster page by following the initial page loading

## 2.5 Refresh Roster Week

Show the following confirmation message

(Are you sure you want to refresh the selected roster week?)

If Yes then refresh the entire roster calendar

## 2.6 Print Roster Week

Friendly print layout page based on the current selected view



## 2.7 Publish Roster Week Shifts

### Publish Roster's Shifts

1 Publish Type
2 Guards / Locations
3 Notifications

Guards can only see shifts you publish  
Send guards a list of their published shifts by email or mobile app

☒ **Publish by guard**  
Display a list of all guards with current roster's week shifts

☐ **Publish by location**  
Display a list of all available locations in the current roster week

Cancel
Back
Next Step

### Select Guards or Locations

1 Publish Type
2 Guards / Locations
3 Notifications

☒ **New / updated shifts**
☐ **Publish all shifts**

Select: [All](#) [None](#) [Inverse](#)

☒ Bondi Beach - ABC
☒ Melbourne DC - XYZ

☒ Jenny Wu
☒ Kate Mosman

Cancel
Back
Next Step

### Publish Shifts

1 Publish Type
2 Guards / Locations
3 Notifications

Send shifts notification via email or mobile app push notification

☒ **Email and Smartphone Push Notification**

☐ No notifications, just mark as published

Cancel
Back
Publish

### Shifts Published

Some shifts were not successfully published.  
You may need to advise these people of their shifts in person.  
It is strongly advised that you keep current and accurate details for all of your Employees.

Employee Name	Reason
Alan Smith	No email address found

Done

There are two publish options

- Publish by Guards
- Publish by Locations

Go to Next Page

If publish by Guards is selected then display all available guards in the current roster week by using the following query:

```
SELECT DISTINCT (wsRstDet.grdID),Guards.gFName,Guards.gLName,
Guards.gMob,Guards.gDvcID FROM (wsRstDet INNER JOIN Guards ON wsRstDet.grdID =
Guards.grdID AND wsRstDet.cIntID = Guards.cIntID) WHERE wsRstDet.cIntID =
$_SESSION['cIntID'] AND wsRstDet.rsID = $_POST['rsID'] ORDER BY
Guards.gFName,Guards.gLName
```

And for publish by Locations, use the

```
SELECT DISTINCT (wsRstDet.sID),wsLoc.sName,wsLoc.sCode FROM (wsRstDet INNER
JOIN wsLoc ON wsRstDet.sID = wsLoc.sID AND wsRstDet.cIntID = wsLoc.cIntID)
WHERE wsRstDet.cIntID = $_SESSION['cIntID'] AND wsRstDet.rsID = $_POST['rsID']
ORDER BY wsLoc.sName
```

User will need to select all required guards and locations

And also select either to publish new and updated shifts or all shifts

Go to Next Page



There are two types of methods to send the selected shifts to guards

- Via Smart Phone push notification or Email
- No notification, just mark as published

### 2.7.1 Publishing selected shifts

If a guard has a mobile app registered then use push notification

If a guard does not have a mobile app registered then send by email if there is an email address

Evaluate guard's shifts published or not by checking `isPub`

Once the publish process is completed, update each shift and set the `isPub` to True

## 3. Roster Shifts

### 3.1 Copy – Paste Shift Process

When a copy is initiated, the cIntID, rsID and shift id (rsDID) should be placed in the clipboard

**Note:** another method is to save all copied shift details into the memory

Copy should only be allowed for a single shift cell

The following process should apply when pasting a copied shift

Check if the copy contents is a valid shift

Check if the roster week is locked by running the following query:

```
SELECT rsRecon FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID']
```

If rsRecon = True then show warning message (The selected roster week has been finalised and reconciled by another user. Ask an administrator to reset the current roster week!) and stop the paste process

Paste can only be applied to a blank cell

#### Guard's on leave check!

Use the following query to check if the new guard is on leave or not

```
SELECT grdID FROM Guards WHERE cIntID = $_SESSION['cIntID'] AND isHide = False  
AND grdID = ['grdID'] AND Not EXISTS (Select lvID FROM grdLve WHERE  
Guards.grdID = grdLve.grdID AND (#['rsDate']# BETWEEN grdLve.lvFrom AND  
grdLve.lvTo))
```

If on leave then stop the paste process

If the copied contents includes IDs only then

Extract cIntID, rsID and rsDID from the copy contents

Use the following query to pull the copied shift details

```
SELECT * FROM wsRstDet WHERE cIntID = $_SESSION ['cIntID'] AND rsID = ['rsID']  
AND rsDID = ['rsDID']
```

If no matching shift found then display error message (No matching shift has been found, either the copied details are incorrect or the original shift has been deleted!)

But if the copied content includes all shift details then

Extract the copied shift details

#### Shift Conflict Check

Check for conflict shift only if the paste process is done on a cell during a location view method

Loop through all guard's shifts by running the following:

```
SELECT rsSTime,rsETime FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND  
rsID = $_POST['rsID'] AND grdID = ['grdID']
```

If the new start time falls between other shifts start and end time then a conflict found

If the new end time falls between other shifts start and end time then a conflict found

If a conflict found then display error message (Guard's shift conflict has been detected, change the start or finish time then try again!) and stop the update process

If copy from one location to another then

Get the customer ID based on the location id by running the following query:

```
SELECT cusID FROM wsLoc WHERE cIntID = $_SESSION['cIntID'] AND sID = ['sID']
```

Check if the copied rate type is found under the new customer only if the location ID is different:

```
SELECT hCode FROM cusRate WHERE cIntID = $_SESSION['cIntID'] AND cusID = ['cusID'] AND hCode = ['hCode']
```

If the rate type is not found then display a popup form for the user to select one of the available rate types

Once all the above validation is passed then insert the copied contents

If the paste is on a cell that belongs to the first row "Unassigned Shifts" then make sure to remove guard ID and set **rsNoGrd** to True

## 3.2 Add New Shift Process

### 3.2.1 Display the new shift popup form and prepare default values

Shift's Status drops down list values are (Pending, Confirmed, Finalised, Declined)

Start time and end time must be according to the client's global format (24 or 12 hours format)

Pre-fill guards drop down list (exclude any inactive and on leave guards)

```
SELECT gFName,gLName FROM Guards WHERE cIntID = $_SESSION['cIntID'] AND isHide = False AND Not EXISTS (Select lvID FROM grdLve WHERE Guards.grdID = grdLve.grdID AND (#[ 'rsDate' ]# BETWEEN grdLve.lvFrom AND grdLve.lvTo)) ORDER BY gFName,gLName
```

**Note:** Drop down list should have the ability for multiple items selection

Repeat On feature:

Add a check box option for the remaining dates within the current roster week

Pre-fill all active locations by running the following query:

```
SELECT sID,sCode,sName FROM wsLoc WHERE cIntID = $_SESSION['cIntID'] AND sHide = False
```

When a location is selected then the rate type should be pre-filled based on the location's parent customer as below:

Extract the location ID `sID` from the location drop down list selected item

Get the customer ID based on the extracted location id by running the following query:

```
SELECT cusID FROM wsLoc WHERE cIntID = $_SESSION['cIntID'] AND sID = ['sID']
```

Run the following query to prefill the rate type drop down list

```
SELECT hCode FROM cusRate WHERE cIntID = $_SESSION['cIntID'] AND cusID = ['cusID']
```

### 3.2.2 Left side contents

Show the guard's name and picture if found

Display the total hours for the selected roster week for the current guard by running:

```
SELECT SUM(rsHours) AS tHrs FROM wsRstDet WHERE cIntID = $_SESSION['cIntID']  
AND rsID = $_POST['rsID'] AND grdID = ['grdID']
```

Display the new shift total hours, (it must be automatically updated when the start time, end time and break values change)

### 3.2.3 Saving New Shift

The following validation rules must be in place before adding the new shift details:

Check if the roster week is locked by another user by running the following queries:

```
SELECT rsRecon FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID']
```

If `rsRecon` = True then show warning message (The selected roster week has been finalised and reconciled by another user. Ask an administrator to reset the current roster week!) and stop the saving shift process

Validate the start time, if incorrect time format then display the error message (Enter a valid start time then try again!)

If end time is not null then validate the end time value, if incorrect then display the error message (Enter a valid end time then try again!)

If end time equal or less than the start time then display the error message (Enter a valid start time/end time then try again!)

If end time is null then set the hours (`rsHours`) to 0 (zero)

If rate type is null then display the error message (Select a rate type then try again!)

If shift's status is null then display the error message (Select a status type then try again!)

#### Shift Conflict Check

Check if the start time and end time have no conflict with other guard's shifts

Loop through all guard's shifts by running the following:

```
SELECT rsSTime,rsETime FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND  
rsID = $_POST['rsID'] AND grdID = ['grdID']
```

If the new start time falls between other shifts start and end time then a conflict found

If the new end time falls between other shifts start and end time then a conflict found

If a conflict found then display error message (Guard's shift conflict has been detected, change the start or finish time then try again!) and stop the update process

If all validation rules passed successfully then insert the new shift

**Note:** If multiple guards are selected then add each guard's shift as a separate shift

If the user checked one of the repeat on dates then the following must be applied by looping through the repeated date checkbox values

Check if there is a conflict shift by using the same procedure as above

If conflict found then do not save the shift on the repeated date BUT do not display any message

If no conflict then insert the same shift details as the original but only replacing the original shift date with the repeated date value

Display the following message (Guard's shift details have been added successfully)

Close the shift details and refresh the current roster week

It is highly recommended to refresh the updated cell only instead of the entire roster calendar week

## 3.2 Edit Shift Process (Single Shift per Guard per Day)

**Edit shift on Wed 02 Jan 2019**

Who is working?

In which location?

Start  Finish  Break (mins)

Rate type

Shift Status

Notes

Actions

### 3.2.1 Display the selected shift

Shift's Status drop down list values are (Pending, Confirmed, Finalised, Declined)

Start time and end time must be according to the client's global format (24 or 12 hours format)

Actions, Delete and Save buttons should be disabled or removed when the shift is locked!

Use the following query to pull the shift's details:

```
SELECT * FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID'] AND rsDID = ['rsDID']
```

Fill the rate types drop down list by using the following query

cusID value is read from the above query

```
SELECT hCode FROM cusRate WHERE cIntID = $_SESSION['cIntID'] AND cusID =  
['cusID']
```

- Guard's name should be locked and cannot be changed
- Location's name should be locked and cannot be changed
- Display start time (**rsSTime**), end time (**rsETime**), hours (**rsHours**) and break (**rsBreak**)
- add rate type (**rsCode**) and shift's status (**rsStatus**)

- Add location name or guard's name based on the selected view
- Add the note (**rsNotes**) and guard's notes (**grdNote**) if found
- If **rsPub** = true then display the published icon
- If **rsLock** = true then display the locked icon and **hide the Save and Delete buttons and Actions menu**

### 3.2.2 Left side contents

Show the guard's name and picture if found

Display the total hours for the selected roster week for the current guard by running:

```
SELECT SUM(rsHours) AS tHrs FROM wsRstDet WHERE cIntID = $_SESSION['cIntID']
AND rsID = $_POST['rsID'] AND grdID = ['grdID']
```

Display the current shift total hours (**rsHours**), (it must be automatically updated when the start time, end time and break values change)

### 3.2.3 Saving Shifts Details

The following validation rules must be in place before updating the current shift details:

Check if the roster week is locked by another user:

```
SELECT rsRecon FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND rsID =
$_POST['rsID']
```

If **rsRecon** = True then show warning message (The selected roster week has been finalised and reconciled by another user. Ask an administrator to reset the current roster week!) and stop the update process

Check if the current shift is locked by another user:

```
SELECT rsLock FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND rsID =
$_POST['rsID'] AND rsDID = ['rsDID']
```

If **rsLock** = True then show warning message (The selected shift has been locked by another user. Ask an administrator to unlock the current shift!) and stop the delete process

Validate the start time (**rsSTime**), if incorrect time format then display the error message (Enter a valid start time then try again!)

If end time (**rsETime**) is not null then validate the end time value, if incorrect then display the error message (Enter a valid end time then try again!)

If end time (**rsETime**) equal or less than the start time (**rsSTime**) then display the error message (Enter a valid start time/end time then try again!)

If end time (**rsETime**) is null then set the hours (**rsHours**) to 0 (zero)

If rate type (**rsCode**) is null then display the error message (Select a rate type then try again!)

If shift's status (**rsStatus**) is null then display the error message (Select a status type then try again!)

#### Shift Conflict Check

Check if the start time and end time have no conflict with other guard's shifts

The conflict process should only occur when the start time or end time have changed

Loop through all guard's shifts excluding the current shift by running the following:

```
SELECT rsSTime,rsETime FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND
rsID = $_POST['rsID'] AND grdID = ['grdID'] AND rsDID <> ['rsDID']
AND rsDID <> ['rsDID']
```

If the new start time falls between other shifts start and end time then a conflict found

If the new end time falls between other shifts start and end time then a conflict found  
If a conflict found then display error message (Guard's shift conflict has been detected, change the start or finish time then try again!) and stop the update process

If start time (rsSTime), end time (rsETime) or break (rsBreak) changed then reset the shift's publish flag (rsPub) to False

If all validation rules passed successfully then update the current shift and display the following message (Guard's shift details have been updated successfully)

Close the shift details and refresh the current roster week

It is highly recommended to refresh the updated cell only instead of the entire roster calendar week

### 3.2.4 Edit Multiple Guard's Shifts in a Single Day

There are rare situations where guards can have 2 shifts in a single day

Display the available shift and ask the user to select a shift

Once a shift is selected then use the same edit procedure for a single shift edit

### 3.2.5 Delete Current Shift

Use the same process under the Delete section

### 3.2.6 Edit Shift's Actions

#### De-assign Shift

Check if roster week and shift are not locked (refer to previous section)

Display a warning message (Are you sure you want to remove the current guard from the current shift?)

If Yes then run the following query

```
UPDATE wsRstDet SET rsNoGrd = True, grdNote = '', rsLock = False, rsPub = False, rsStatus = 'Pending', grdID = 0 WHERE cIntID = $_SESSION['cIntID'] AND rsID = $_POST['rsID'] AND rsDID = ['rsDID']
```

Display successful message (Current shift has been successfully de-assigned)

Exit and Refresh the current roster week

#### Re-assign Shift

Check if roster week and shift are not locked (refer to previous section)

Display a warning message (Are you sure you want to re-assign the current shift to another guard?)

If Yes then display a list of available guards excluding the current guard and guards on leave

```
SELECT gFName,gLName FROM Guards WHERE cIntID = $_SESSION['cIntID'] AND grdID <> ['grdID'] AND isHide = False AND Not EXISTS (Select lvID FROM grdLve WHERE Guards.grdID = grdLve.grdID AND (#['rsDate']# BETWEEN grdLve.lvFrom AND grdLve.lvTo)) ORDER BY gFName,gLName
```

User must select a guard from the list

#### Shift Conflict Check

Check if there is a shift conflict for the new guard (refer to previous section)

Run the following query

```
UPDATE wsRstDet SET rsNoGrd = True, rsLock = False, rsPub = False, rsStatus = 'Pending', grdID = ['newGrdID'] WHERE cIntID = $_SESSION['cIntID'] AND rsID = $_POST['rsID'] AND rsDID = ['rsDID']
```

Display successful message (Current shift has been successfully re-assigned)



Exit and Refresh the current roster week

### Lock Shift

Check if roster week and shift are not locked (refer to previous section)

Display a warning message (Are you sure you want to lock the current shift?)

If Yes then Run the following query

```
UPDATE wsRstDet SET rsLock = True WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID'] AND rsDID = ['rsDID']
```

Display successful message (Current shift has been successfully locked)

Exit and Refresh the current roster week

### Add Leave

Check if the guard has a leave record for the current date

If a leave record found then display a message (A leave record has been already added for the current shift's date?)

Display the leave form

Validate Leave from date and leave to dates

Validate the leave type (if not unpaid leave and the current guard is an employee (isEmp) then the user must enter the total leave hours

Save the leave record and display a successful message

## 3.3 Delete Shift Process

Check if the roster week is locked by running the following query

```
SELECT rsRecon FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID']
```

If rsRecon = True then show warning message (The selected roster week has been finalised and reconciled. Ask an administrator to reset the current roster week!) and stop the delete process

If not locked, then check if the shift is locked by running the following query

```
SELECT rsLock FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID'] AND rsDID = ['rsDID']
```

If rsLock = True then show warning message (The selected shift has been locked. Ask an administrator to unlock the current shift!) and stop the delete process

If not locked then display the following confirmation message

(Are you sure you want to delete the selected shift?)

If Yes then run the following query to delete the selected shift

```
DELETE FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID'] AND rsDID = ['rsDID']
```

Refresh the current display roster week by removing the current deleted cell contents

It is highly recommended to remove the deleted cell only instead of refreshing the entire roster calendar week

## 4. Shifts Authorisation Actions

Shift authorisation is a process where the roster staff member can review and process the submitted details via a mobile app by a guard

All pending authorisation records should be displayed as an alert in the footer section of the roster calendar as shown in the main layout

### 4.1 Status Report

This report should show all guards that submitted their acceptance or decline published shifts

When a guard submits roster shifts, it connects via an API to store the information in a temp table called **wsRstGrds**

Use the following query to check if there are pending shifts

```
SELECT Count(*) FROM wsRstGrds WHERE cIntID = $_SESSION['cIntID'] AND rsID = $_POST['rsID']
```

If pending shifts are found then add the red + symbol to the Status Report link in the bottom of the calendar as an indication to the roster admin that there are pending reports

#### 4.1.1 Authorising Status Report

When there are pending shifts, a roster admin can click on the link and should have new page which lists all available pending shifts.

This page should have a search option by Guard, locations or All pending shifts

Roster admin can then select the required shifts to authorise

By clicking on Authorise button, the following must occur:

Update the selected shift **rsStatus** from **wsRstGrds** and matching the same shift on the original roster table **wsRstDet**

Once the update is successful then the pending shift from the temp table **wsRstGrds** should be removed

### 4.2 Weekly Report

This report should show all guards that submitted their weekly report

When a guard submits a weekly report, it connects via an API to store the information in a temp table called **wsRstReps**

If pending weekly reports are found then add the red + symbol to the Weekly Report link in the bottom of the calendar as an indication to the roster admin that there are pending reports

#### 4.2.1 Authorising Weekly Report

When there are pending weekly reports, a roster admin can click on the link and should have new page which lists all available pending reports.

This page should have a search option by Guard

The report should show two panels or sections

First section should list all current shifts in the roster by using the following query:

```
SELECT * FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND rsID = $_POST['rsID'] AND grdID = ['grdID']
```

Second section should list all available pending options by using the following query:

```
SELECT * FROM wsRstReps WHERE cIntID = $_SESSION['cIntID'] AND rsID = $_POST['rsID'] AND grdID = ['grdID']
```

Roster admin manually review and compare the two reports

If new shift is submitted by a guard which was not listed in the roster then the roster admin should be able to click on a button to add it to the main roster

By clicking on Authorise button, the following must occur:

All pending weekly reports from the temp table `wsRstGrds` should be removed

## 4.3 Leave Request

This report should show all guards that submitted a leave request

When a guard submits a leave request, it connects via an API to store the information in a temp table called `grdTmPLve`

If pending leave requests are found then add the red + symbol to the Leave Request link in the bottom of the calendar as an indication to the roster admin that there are pending reports

### 4.3.1 Authorising Leave Request

When there are pending Leave Requests, a roster admin can click on the link and should have new page which lists all available pending leave requests.

This page should have a search option by Guard or All Leave Requests

Use the following query to list pending requests

```
SELECT grdTemLve.*,Guards.gFName,Guards.gLName FROM (grdTemLve INNER JOIN
Guards ON grdTemLve.grdID = Guards.grdID AND grdTemLve.cIntID = Guards.cIntID)
WHERE grdTemLve.cIntID = $_SESSION['cIntID'] AND grdTemLve.gStatus = 'Pending'
Guards.gFName,Guards.gLName
```

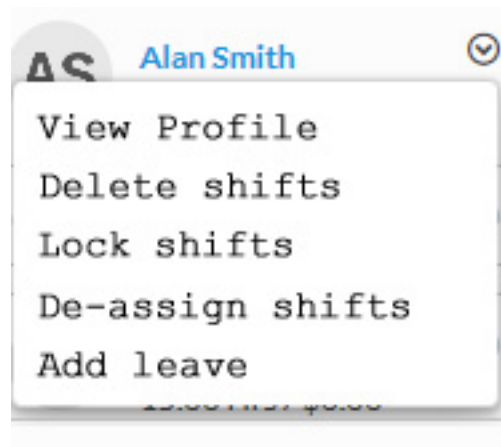
Roster admin should have the ability to authorise a leave or decline

When the leave is authorised, then the following should occur:

- Update the main leave table `grdLve` with the authorised leave details
- Flag the temp leave table `grdTmPLve` as Accepted

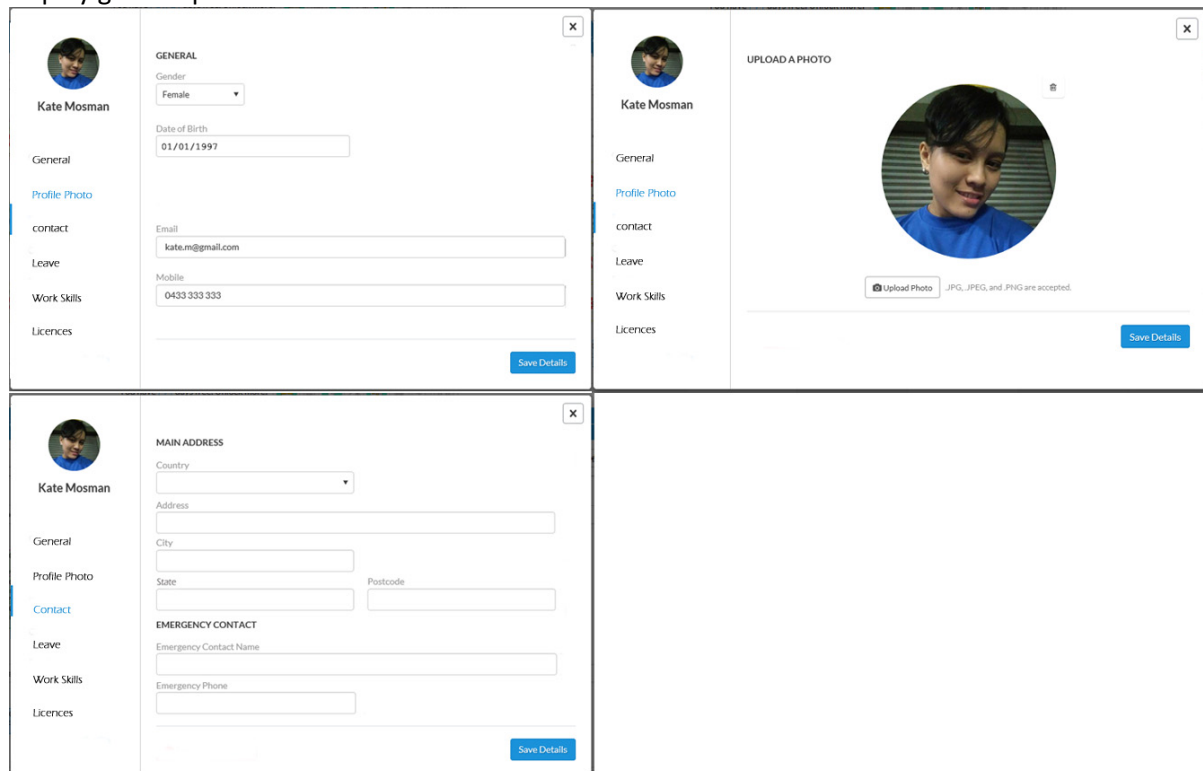
When the leave is rejected then flag the temp leave record as rejected

## 5. Guard's Menu



### 5.1 View Profile

Display guard's profile details



View and update basic details and upload guard's picture

Use the following queries to read guard's information

```
SELECT * FROM Guards WHERE cIntID = $_SESSION['cIntID'] AND grdID = ['grdID']
```

Read Leave records

```
SELECT * FROM grdLve WHERE cIntID = $_SESSION['cIntID'] AND grdID = ['grdID']
```

Read Work Skills

```
SELECT * FROM grdSkls WHERE cIntID = $_SESSION['cIntID'] AND grdID = ['grdID']
```

Read Licences

```
SELECT * FROM grdLics WHERE cIntID = $_SESSION['cIntID'] AND grdID = ['grdID']
```

## 5.2 Delete Guard's Shifts

Check if the roster week is locked by running the following query

```
SELECT rsRecon FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID']
```

If `rsRecon = True` then show warning message (The selected roster week has been finalised and reconciled. Ask an administrator to reset the current roster week!) and stop the delete process

If not locked then display the following confirmation message

(Are you sure you want to delete the selected guard's shifts?)

If Yes then run the following queries to delete the selected shift

```
DELETE FROM wsRstDet WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID'] AND grdID = ['grdID']  
DELETE FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID'] AND grdID = ['grdID']  
DELETE FROM wsRstReps WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID'] AND grdID = ['grdID']  
DELETE FROM wsRstGrds WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID'] AND grdID = ['grdID']
```

Refresh roster calendar by removing the selected guard

## 5.3 Lock Shifts

Check if the roster week is locked by running the following query

```
SELECT rsRecon FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND rsID =  
$_POST['rsID']
```

If `rsRecon = True` then show warning message (The selected roster week has been finalised and reconciled. Ask an administrator to reset the current roster week!) and stop the locking process

Check if guard's shifts are locked

If locked then show the following message to reset

(Are you sure you want to reset the selected guard's shifts?)

If Yes then set `rsLock` to true

If not locked then show the following message:

(Are you sure you want to lock and reconcile the selected guard's shifts?)

If Yes then set `rsLock` to true

## 5.4 De-assign Shifts:

Check if the roster week is locked by running the following query

```
SELECT rsRecon FROM wsRoster WHERE cIntID = $_SESSION['cIntID'] AND rsID = $_POST['rsID']
```

If `rsRecon = True` then show warning message (The selected roster week has been finalised and reconciled. Ask an administrator to reset the current roster week!) and stop the de-assigning process

(Are you sure you want to de-assign the selected guard's shifts?)

If Yes then remove guard ID and set `rsNoGrd` to True and `rsStatus` to Pending (P)

Once all shifts are de-assigned then remove the current guard

## 5.5 Add Leave

Guards with approved leaves should not be assigned any shifts during their leave period

The screenshot shows a web form titled "Kate Mosman - Leave". The form includes the following elements:

- Leave request from:** A date input field containing "Thu 03/01/19".
- Leave request to:** A date input field containing "Thu 03/01/19".
- Leave Type:** A dropdown menu with "Unpaid Leave" selected.
- Total Hours:** A numeric input field containing "7.00", followed by the unit "hrs".
- Description:** A large text area for additional details.
- Buttons:** "Cancel" and "Add & Approve" buttons at the bottom right.

Check if the selected guard is active or not, if inactive then display the following and exit

(The current guard is inactive, leave records can only be added to active guards!)

Check if there is an existing record by matching Leave From, Leave to and Leave type

If a matching record found then display the following message and exit:

(An existing leave record found matching the require leave period, change the leave period or use the leave records to update the required leave details!)

All Ok, then insert the new leave details into `grdLve` and display a successful message