Salary Management System

**Run Payroll**

This module will populate from **Master table** employee ID and Names and will give option to fill

Non static values for each like

* *Work Days*
* *Others*
* *Arrears*
* *On Call Allowance*
* *Laptop Allowance*

*Arrear Deduction* should be auto calculated.

Based on these, entries will be created in **Payroll** table.

**Master**

> Ensure that Master has compensation start and end date.

> Also Current flag must be there.

While running the payroll master can be referred with current flag set.

An auto functionality to disable the existing master record while making new of same user applicable.

**Retrospective calculation**

This must be done after using run payroll feature with the current package.

Then Retro page will allow us to run retro adjustment from the old package.

It will ask for a start date from and should apply the differential gross of previous months to old gross as arrears in new sal package. Also it will be taking care pf and other deductions individually.

**YTD Tables:**

This is required and can be updated when payroll is inserted

**Run Payroll :**

Using Open payroll calculate and enter values in payroll table.

Ensure that offset is available for all calculations

After insertions, final adjustments be allowed for the person.

**Salary slip ( earnings and deductions ) :PF Slip ( deducted and contributed ): YTD Slip (earnings & deductions)**

Generate on web and export to PDF

**Validations in the last:**

**Make an Ajax data fetch model in Json format.**

**Extra files:**

|  |  |
| --- | --- |
| File Name | Description |
| data.php | **gets display data** |
| event.html | **page to run events** |
| db.php | **Created for db connection instead of auth.php** |
| lib/insert\_ajax.php | **replaces insert.php** |
| data\_sql.php | **replaces** |
| chart.html | **generates chart from Ajax using Flot** |

**General Design considerations:**

**Javascript events**

### Excel Import

Excel import works in following way

Excelimport.php refers PHPOffice library from the parent folder and calls readexcel.php

#### readexcel.php

This file is written as a main calling file to invoke PHPOffice library excel classes.

Also it has code to parse the excel based on import done and show it in editable mode using form <input> tags.

#### Configuring Excel

The excel import can be configured in IO Configuration

We can define the template of the file there to be imported and use it as it is.

The template will decide the type of file shown.

This has Three Sections:

Print Header

Properties are as here

|  |  |
| --- | --- |
| Attribute | Description |
| rhdr | Rows from Starting |

Header Row

Data Rows

! Footer is generally ignored to avoid any confusion with Data.

### General

Summary and Details section

On click of Summary , details for a row should be shown.

### /lib/Menu.php

This has been modified so that ‘Style” field can be set to 0 where we want link to be shown.

Else it can be set to 1 or any other number if link doesn’t need to be provided.

### Transaction

This table will record the current running payroll

### Payroll Status

If it is open, then only a payroll can be run and finally completed.

Open Payroll

## Steps to Run the Payroll

### Dashboard

* Dashboard should show all open payrolls. In case no one is open Open Payroll button should be enabled.
* Open Payroll for a month – Select Month and Year and click Open.

#### Transaction

* + This should create a entry in a transaction table with payroll status open
  + Fields Month, Year , Status( Open, Closed), File Uploaded, RunDate, Errors
    - This table should contain history of all payrolls open till now.
    - Only if all previous are closed then a new should be created, else it will give error.
* This new record will show up on transaction and will have file upload functionality enabled.
* Upload a file with Name, Empid , month days , quarterdays for the people.
  + An option to modify will be given before doing a final import of the file.
  + This will go in Payroll Table

### Run Payroll

This Table has following fields (Name, Empid , month days , quarterdays).

Once uploaded, the records will hv transaction ID

Then payroll table will have calculations running on it.

## Workflow Design

Design workflow engine

* 1. Create table to Store Workflow name, ID and type(client or server)
  2. If it is client side, it will use Ajax to call php code on trigger of Events in Javascript.
  3. Create Workflow table to map the form name/tblID ( config is table which stores tblid of forms)

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int(10) | primary key |
| name | varchar(100) | Workflow Name |
| type | int(1) | 0 stands for client side and 1 for server side |
| tblid | int(10) | ID of table to be associated from config table |
| num | int(3) | Order in which this workflow will get precedence |
| fieldname | int(10) | Generally should be a button name used to submit the form |
| qual | varchar(255) | Criteria on which code block should run |

* 1. Create a table to store action

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int(10) | primary key |
| wkflid | int(10) | ID of workflow form workflow table |
| seq | int(2) | sequence in which this getValues will run within wkflid |
| type | int(1) | 0= get values  1= put values |
| tblid | int(10) | table to which values will be insert/updated or read from |
| qual | varchar(255) | Criteria on which values will be inserted |
| modn | int(1) | If populated its decoded as below  0 stands for modify all  1 stands for modify first  2 stands for modify last  Else new record will be created |

* 1. Create table action\_map to store values mapping

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int(10) | primary key |
| actionid | int(10) | id from action table |
| cfieldid | int(10) | Current form fieldID |
| sfieldid | int(10) |  |
| svalue | varchar(255) | add formula or qual from the source table or add formula or qual from the source table to push to target |

* 1. In php code write a function to do the following
     1. Select tblid for th current form from the config.
     2. Select values from workflow table and capture all the workflow ids where current tblid.
     3. Select Values from action table based on the wkflid from step 2 and order by seq.
     4. Now run the loop 1 for each workflow selected
        1. Open another loop 2 for the each actionid and do as below
           1. Set $condition=qual, and check if($condition) then progress to b else exit loop.
           2. Select all from action\_map table where action Id is current

Prepare a sql for selec t and update is the action is get else prepare the sql for update/insert if the action is put

Open loop and prepare the fname( field names) and fval (field Values)

Join the qual in sql string

Execute the sql as required

* + 1. Function should end up doing the job

## Single record view

Arrange fields in table in such a manner that single record view be there..