# Khởi tạo lương mặc định:

* Phương án 1: Nhập tháng cần khởi tạo => hệ thống tự động tính lương cho tất cả nhân viên (chưa bị xóa) trong tháng đó. Các dữ liệu chấm lương như ngày nghỉ, phụ cấp … mặc định bằng 0. Dữ liệu được lưu lại, bao gồm cả ngày giờ và người khởi tạo. Trạng thái: chưa duyệt.
* Phương án 2: Đặt lịch cho hệ thống tự động tính lương cho tất cả nhân viên (chưa bị xóa) trong tháng. Các dữ liệu chấm lương như ngày nghỉ, phụ cấp … mặc định bằng 0. Dữ liệu được lưu lại, bao gồm cả ngày giờ và người khởi tạo (system). Trạng thái: chưa duyệt. (sử dụng store procedure)

**Note: Cách nhập tháng khởi tạo**

**B1: Tạo box loại date để nhập ngày tháng (chọn unset value là zero, pattern là MM-yyyy)**

**B2: Chỉnh filter property của live variable. Bind trường cần filter với box tạo ở B1**

# Chấm lương lần 1:

* Nhập tháng, năm cần chấm lương =>hiển thị những bản ghi trong tháng, năm đó còn trong trạng thái chưa được duyệt (chỉ có 2 trạng thái: duyệt và chưa duyệt)
* Có thể xóa các bản ghi này. Dữ liệu đẩy sang bảng xóa, lưu lại ngày giờ và người xóa
* Chấm lương: Nhập số ngày nghỉ, hệ thống tự tính ra lương. Dữ liệu được lưu lại, bao gồm cả ngày giờ và người chấm. Trạng thái: chưa duyệt.

**Note: Khi lấy dữ liệu từ live variable, cần lưu ý case sensitive**

**Note: Cách truyền dữ liệu từ DB vào javascript:**

**B1: Tạo livevariable lấy dữ liệu từ bảng cần thiết**

**B2: Lấy dữ liệu từ row cần thiết. VD như getItem(0) – 0 là row đầu tiên**

**B3: Lấy trường cần thiết từ row ở B2. VD như getValue(“tên trường”)**

**Note: Cách log javascript**

**Sử dụng console.log(). Bật chế độ test, xem phần event log.**

# Chấm lương thủ công:

* Nhập đầy đủ thông tin để chấm lương. Dữ liệu sẽ được lưu lại bao gồm ngày giờ và người chấm. Trạng thái: chưa duyệt

**Note: Các editor chứa thông tin static của nhân viên được bind với lookup editor của nhân viên**

**Sử dụng java service: (xem form java service)**

* **Khai báo service variable, action là java service**
* **Bind các đầu vào cho service variable**
* **Khai báo các event onResult (required), onSuccess, onError cho service varriable**
* **Cần bind result của service variable với các editor trên liveform. Cách access result của service variable:**
  + **Var result = inSender.getItem(“0”); -- lấy row đầu tiên trong kết quả trả về (trong trường hợp output là dạng non object thì *inSender.getData().dataValue*)**
  + **Result.data.[tên trường]**
* **Nếu cần save data thì gọi câu lệnh saveData của live Form. Cần gọi ngay sau khi bind result (trong cùng 1 script)**

# Duyệt lương lần 1:

* Nhập tháng cần chấm lương =>hiển thị những bản ghi trong tháng đó còn trong trạng thái chưa được duyệt.
* Duyệt: Chuyển dữ liệu sang trạng thái đã duyệt hoặc từ chối. Dữ liệu được lưu lại, bao gồm cả ngày giờ và người chấm.

# Chấm lương lần 2:

* Nhập tháng cần chấm lương =>hiển thị những bản ghi trong tháng đó còn trong trạng thái chưa được duyệt / từ chối.
* Có thể xóa các bản ghi này
* Chấm lương: …

# Duyệt lương lần 2:

* Nhập tháng cần chấm lương =>hiển thị những bản ghi trong tháng đó còn trong trạng thái chưa được duyệt.
* Duyệt: Chuyển dữ liệu sang trạng thái đã duyệt hoặc từ chối. Dữ liệu được lưu lại, bao gồm cả ngày giờ và người chấm.
* Sau khi chuyển sang trạng thái đã duyệt, hệ thống tự động tính toán gộp lương 2 lần và tính thuế

# Điều chỉnh sai sót:

* Đối với những dữ liệu sai mà chưa duyệt, có thể vào chấm lại hoặc xóa.
* Đối với những dữ liệu duyệt rồi mới phát hiện sai, vào màn hình này tìm kiếm theo tên nhân viên và kỳ lương. Chuyển trạng thái sang chưa duyệt.

# Báo cáo:

* Báo cáo lương lần 1. Đầu vào: Kỳ lương
* Báo cáo lương lần 2. Đầu vào: Kỳ lương
* Báo cáo thu nhập tổng. Đầu vào: Kỳ lương
* Báo cáo thu nhập cả năm. Đầu vào: Năm

# Danh sách nhân viên:

* Xem thông tin nhân viên (chỉ hiện các nhân viên chưa xóa – có trạng thái isdeleted là 0)
* Cập nhật thông tin nhân viên – lưu lại thời gian, người cập nhật
* Thêm mới nhân viên – lưu lại thời gian, người cập nhật
* Xóa nhân viên (chuyển trạng thái là xóa, trạng thái isdeleted là 1) – lưu lại thời gian, người xóa

**Note: Cách lưu thời gian, người cập nhật**

**Tạo javascript khi có event thêm, sửa, xóa: trong script này set datavalue của trường thời gian, người cập nhật**

# Danh sách nhân viên đã xóa:

* Xem các nhân viên đã xóa
* Cập nhật lại thành chưa xóa, trạng thái isdeleted là 0 – lưu lại thời gian, người khôi phục

# Danh sách phòng ban

* Xem thông tin phòng ban
* Cập nhật thông tin phòng ban – lưu lại thời gian, người cập nhật
* Thêm mới phòng ban – lưu lại thời gian, người cập nhật
* Xóa phòng ban (chuyển sang bảng xóa). Kiểm tra thông tin nhân viên, nếu có nhân viên thuộc phòng ban này thì update lại mã phòng ban là null (taọ trigger)– lưu lại thời gian, người xóa

# Mã ngạch lương

* Xem thông tin ngạch lương
* Cập nhật thông tin ngạch lương – lưu lại thời gian, người cập nhật
* Thêm mới ngạch lương – lưu lại thời gian, người cập nhật
* Xóa ngạch lương (chuyển sang bảng xóa). Kiểm tra thông tin nhân viên, nếu có nhân viên thuộc mã ngạch lương này thì update lại mã ngạch lương là null (tạo trigger)– lưu lại thời gian, người xóa

# Tham số hệ thống:

* Thay đổi tham số hệ thống (không được phép xóa, thêm bản ghi) – lưu lại thời gian, người cập nhật

# Bảo mật chương trình:

* Mã hóa dữ liệu (mật khẩu người dùng)
* …

# Quản trị người dùng:

* Thêm mới người dùng, lưu thời gian, người tạo
* Sửa thông tin người dùng (sửa mật khẩu), lưu thời gian, người sửa
* Xóa người dùng

# Phân quyền:

* Phân quyền người dùng
* Xóa phân quyền người dùng

# PL1. Cách tạo thành phần chung cho tất cả các site:

1. Tạo menu, header, footer … ở main page.
2. Tạo các page khác chỉ có thành phần lõi, không bao gồm menu, header, footer …
3. Tạo 1 page container trong main page.
4. Tạo event khi click menu là gotopagecontainerpage, sau đó chọn page tương ứng.

# PL2. Hash password

## PL2.1. Config lại wavemaker project

1. Download spring-security-crypto-\*.jar into /lib  
2. Created PasswordEncoder class in /src  
3. Modify project-security.xml

Created a class in /src, the following code:

package com.abc.wm.security;  
  
import org.acegisecurity.providers.encoding.PasswordEncoder;  
import org.springframework.dao.DataAccessException;  
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;  
  
public class BcryptPasswordEncoder implements PasswordEncoder {  
  
public static final int DEFAULT\_BCRYPT\_STRENGTH = 12;  
  
protected BCryptPasswordEncoder bce = null;  
  
protected int strength = DEFAULT\_BCRYPT\_STRENGTH;  
  
public BcryptPasswordEncoder() {  
this(DEFAULT\_BCRYPT\_STRENGTH);  
}  
  
public BcryptPasswordEncoder(int strength) {  
this.strength = strength;  
bce = new BCryptPasswordEncoder(strength);  
}  
  
@Override  
public String encodePassword(String pwd, Object salt) throws DataAccessException {  
return bce.encode(pwd);  
}  
  
@Override  
public boolean isPasswordValid(String encPwd, String rawPwd, Object salt) throws DataAccessException {  
return bce.matches(rawPwd, encPwd);  
}  
  
public void setStrength(int strength) {  
this.strength = strength;  
bce = new BCryptPasswordEncoder(strength);  
}  
}

Then I modified the project-security.xml, adding the bcryptPasswordEncoder bean and passwordEncoder property to the daoAuthenticationProvider bean.

    <bean class="org.acegisecurity.providers.ProviderManager" id="authenticationManager">  
        <property name="providers">  
            <list>  
                <ref bean="daoAuthenticationProvider"/>  
                <ref bean="anonymousAuthenticationProvider"/>  
            </list>  
        </property>  
    </bean>  
    <bean class="com.abc.wm.security.BcryptPasswordEncoder" id="bcryptPasswordEncoder" >  
        <property name="strength" value="12"/>  
    </bean>  
    <bean class="org.acegisecurity.providers.dao.DaoAuthenticationProvider" id="daoAuthenticationProvider">  
        <property name="userDetailsService">  
            <ref bean="jdbcDaoImpl"/>  
        </property>  
        <property name="passwordEncoder">  
            <ref bean="bcryptPasswordEncoder"/>  
        </property>  
    </bean>

After that the code works.

## PL2.2. Sử dụng java service

Ngoài ra, các form đăng ký user, reset mật khẩu, etc có thể sử dụng thư viện Bcrypt để hash passwor như sau:

String pwd = "trankynam";

String salt = BCrypt.*gensalt*(12);

String encPassword = BCrypt.*hashpw*(pwd, salt);