

**VIETTEL PERU**

**<DEPARTMENT: >**

**<Project name>**

**DETAIL DESIGN**

**Project code:**

**Document code:**

**<Lima, Date>**

**TRACING TABLE**

\*A – Create New, M – Modify, D – Delete

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Day**  **change** | **Position**  **change** | **A\***  **M, EASY** | **Source** | **DISCIPLINE** | **Description change** | **Note** |
| October 15, 2010 | Request change number 1 | A | Dispatch XYZ | PTC, PKH | Content requested to change |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**PAGE**

Founder: <Date>

<Title>

Reviewer: <Date>

<Title>

Reviewer: <Date>

<Title>

Approved by: <Date>

<Title>

TABLE OF CONTENTS

[1.](#_Toc315976226)  [INTRODUCTION](#_Toc315976226)  [5](#_Toc315976226)

[1.1](#_Toc315976227)  [Purpose](#_Toc315976227)  [5](#_Toc315976227)

[1.2](#_Toc315976228)  [Range](#_Toc315976228)  [5](#_Toc315976228)

[1.3](#_Toc315976229)  [Concepts and terms](#_Toc315976229)  [5](#_Toc315976229)

[1.4](#_Toc315976230)  [References](#_Toc315976230)  [5](#_Toc315976230)

[1.5](#_Toc315976231)  [Document Description](#_Toc315976231)  [5](#_Toc315976231)

[2.](#_Toc315976232)  [SOLUTION OVERVIEW](#_Toc315976232)  [6](#_Toc315976232)

[2.1](#_Toc315976233)  [Function overview](#_Toc315976233)  [6](#_Toc315976233)

[2.2 Architectural model of](#_Toc315976234)  [6 . components](#_Toc315976234)

[3.](#_Toc315976235)  [DETAILED DESIGN](#_Toc315976235)  [7](#_Toc315976235)

[3.1](#_Toc315976236)  [POLICY MANAGEMENT SYSTEM](#_Toc315976236)  [7](#_Toc315976236)

[3.1.1](#_Toc315976237)  [Resolve the](#_Toc315976237)  **[Error! Bookmark not defined.](#_Toc315976237)**

[4.](#_Toc315976238)  [GENERAL DESIGN AND REUSABLE](#_Toc315976238)  [31](#_Toc315976238)

# INTRODUCE

*[The introduction to the system architecture document should provide an overview of the document including the purpose of the document, concepts, terminology, and related references.]*

## Purpose

This document provides an overview of the system through a number of different architectural models to describe the system from different perspectives. This document is expected to document and communicate important decisions about the system's architecture.

[This section defines the role or purpose of the system architecture document and briefly introduces the document's layout. It should be specified who will read this document and how it will be used.]

## Limit

[A brief description of what does the System Architecture document serve? What are the implications or effects of this document?]

## Concepts and terms

[This section will provide definitions of all the concepts, terms, etc. used in the System Architecture document.]

## References

[The section lists a complete list of all external references. Each document is identified by its title, author, and issue date, and must also indicate whether the work is available or for reference only. If appropriate, specify the report number, title of journal, and issuer of the document.]

## Document Description

[This section will introduce the rest, which are not included in the above four sections, of the System Architecture document and also introduce the layout of the System Architecture document.]

# SOLUTION OVERVIEW

## Function overview



## model with other system/function module

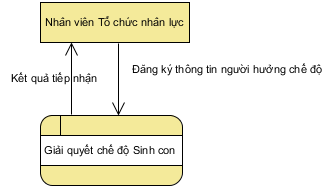
N/A

# DETAILED DESIGN

## POLICY MANAGEMENT SYSTEM ONLY

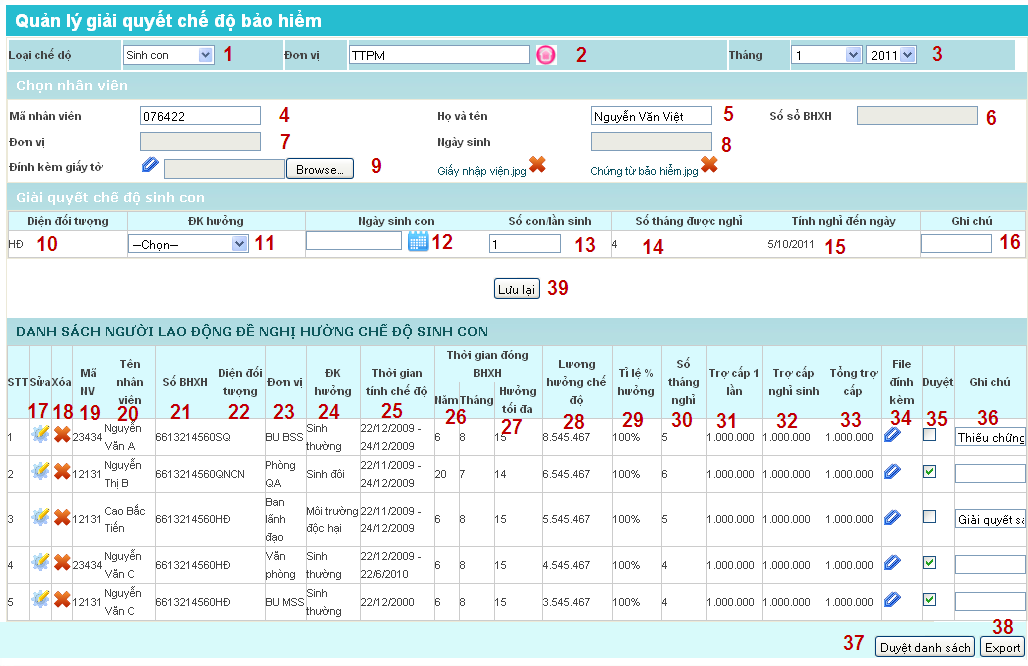
### Solve Childbirth mode

#### Function General Information



Employees of the Human Resources Organization of a corporation, a vertical labor organization are the subjects that use this function. The function allows updating information about the birth control regime for each employee at the unit.

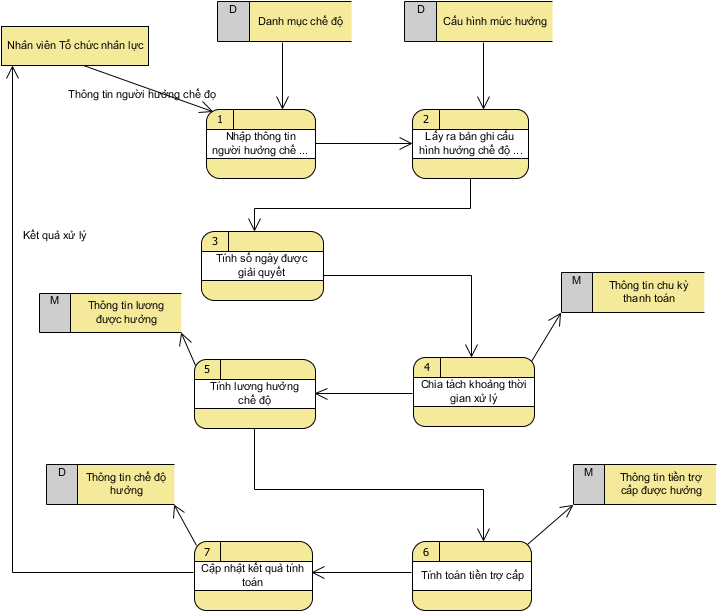
#### Screen



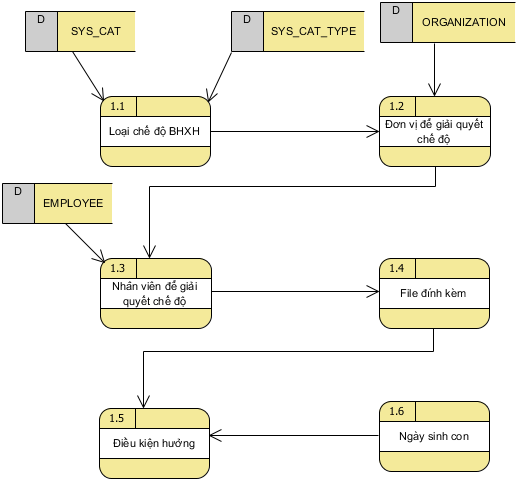
#### Detailed description of ingredients

| **STT** | **Name** | **Datatypes**  **[Data Length]** | **Input/Output** | **Initialization value** | **Description** (Mapping with database if available) |
| --- | --- | --- | --- | --- | --- |
|  | cbxModeType | Combo box |  | Sick | * The type of regime to deal with the policy includes: Sickness, Maternity sickness, childbirth * The data in this combobox is loaded from the SYS\_CAT table with SYS\_CAT\_TYPE = 'Type of social insurance' * Display name in combobox is SYS\_CAT.NAME |
|  | popupOrgChoose | Popup |  | N/A | * This is the Unit Selection Popup that is shared to select the mode settlement unit * Only select units within the scope of user permissions * Select the units that are in effect |
|  | cbxMonthCaculate | Combo box |  | Month/year at the present time in | * Default display is month/year at the time of settlement * Inputted by the user, the data is uploaded in the com * bobox month from 1-12, data loaded in combobox year from [Current Year – 5] to [Current Year + 5] |
|  | txtEmployeeCode | Textbox | INPUT | NULL | * The information field used to search for employees 🡪Datapicker searches for employees is located here * The onclick event handler reference is below. |
|  | txtEmployeeFullName \_ | Textbox | OUTPUT | NULL | * Employee's full name, default display when selecting employee from Datapicker * Taken from the table EMPLOYEE.FULLNAME |
|  | txtInsuranceNumber | Textbox | OUTPUT | NULL | * Number of employee's social insurance book, default display when selecting employee from Datapicker * Taken from the INSURANCE\_SOCIAL\_BASE table. INSURANCE\_NUMBER |
|  | txtOrgName | Textbox | OUTPUT | NULL | * Display the name of the unit containing the employee * The unit name is displayed from the low 🡪to the high unit. Example: Solution Department 🡪MSS 🡪TTPM 🡪Viettel Group * Taken from table ORGANIZATION.NAME |
|  | txtDateOfBirth | Textbox | OUTPUT | NULL | * Employee's date of birth, default display when selecting employee from Datapicker * Taken from table EMPLOYEE.BIRTH\_OF\_DATE |
|  | txtAttachment | Textbox | INPUT | NULL | * File attached * The onclick event handler reference is below. |
|  | lblEmpType | Label | OUTPUT | NULL | * Employee's object name * Mapping corresponds to the EMP\_TYPE table. NAME * Default shows the Employee's current EMP\_TYPE\_PROCESS Object Process (Is the process with EMP\_TYPE\_PROCESS. effective\_start\_date < cbxMonthCaculate.value < EMP\_TYPE\_PROCESS. effective\_end\_date |
|  | cbxCondition | Checkbox | INPUT | NULL | * Employee benefit conditions * The data at the combo-box is taken from the INSURANCE\_CONDITION table (displayed by the NAME field) * However, this data satisfies the condition that INSURANCE\_CONDITION.MODE\_TYPE\_ID = Eligibility condition was selected in the combobox **cbxModeType** above. * When saving mapping data corresponding to INSURANCE\_PAYMENT. INSURANCE\_CONDITION\_ID |
|  | dateBirthOfChild | Textbox | INPUT | NULL | * Employee's date of birth, choose from Datapicker * When saving mapping data corresponding to INSURANCE\_PAYMENT. CHILD\_DATE\_OF\_BIRTH CARD |
|  | txtNumberBirth | Textbox | INPUT | first | * Number of children per birth * When saving mapping data corresponding to INSURANCE\_PAYMENT. NUMBER\_CHILDS\_PER\_ONE |
|  | lblNumberMonth | Label | OUTPUT | NULL | * Number of months of maternity leave, uploaded from the database based on the employee's object configuration * This data is displayed due to the 🡪Load calculation from the INSURANCE\_BENEFIT\_CONFIG table with the corresponding configuration record and is calculated by the following formula: * Number of months of maternity leave = INSURANCE\_BENEFIT\_CONFIG. MAX\_NON\_WORKING\_DAY + (txtNumberBirth.value – 1) / INSURANCE\_BENEFIT\_CONFIG. PROGRESSIVE\_STEP \* INSURANCE\_BENEFIT\_CONFIG .PROGRESSIVE\_ALLOWANCE |
|  | lblUptoDate | Label | OUTPUT | NULL | * Calculation of leave to date (after the childbearing period) |
|  | txtNotesUp | Textbox | INPUT | NULL | * User-entered notes * When saving mapping data corresponding to INSURANCE\_PAYMENT.NOTE |
|  | btnEditData | Button |  | ENABLED | * This button to edit the data in the list of children entitled to the maternity regime * Refer to event handler below |
|  | btnDeleteData | Button |  | ENABLED | * This button to delete data in the list of entitled to childbirth mode * Refer to event handler below |
|  | lblEmployeeCode | Label Column |  | NULL | * Employee code column label in grid * The data in this column mapping corresponds to EMPLOYEE.EMPLOYEE\_CODE |
|  | lblEmployeeFullName | Label Column |  | NULL | * Label the column of the employee's first and last name in the grid * The data in this column mapping corresponds to EMPLOYEE.FULL\_NAME |
|  | lblInsuranceNumber | Label Column |  | NULL | * Label number of employee's social insurance book in grid * The data in this column mapping corresponds to INSURANCE\_SOCIAL\_BASE.INSURANCE\_NUMBER |
|  | lblEmpType | Label Column |  | NULL | * Employee object name label in grid * The data in this column mapping corresponds to INSURANCE\_SOCIAL\_BASE.INSURANCE\_NUMBER |
|  | lblOrgName | Label Column |  | NULL | * Label the unit name in grid |
|  | lblCondition | Label Column |  | NULL | * Label the name of the benefit condition in the grid * The data in this column mapping corresponds to INSURANCE\_CONDITION.NAME |
|  | lblTimeToInsurance | Label Column |  | NULL | * Time stamp of social insurance benefits * The data in this column mapping corresponds to INSURANCE\_PAYMENT. START\_ACTUAL\_DATE-INSURANCE\_PAYMENT.END\_ACTUAL\_DATE |
|  | lblTimeMonthYear | Label Column |  | NULL | * Time stamp of social insurance payment * The data in this column mapping corresponds to INSURANCE\_PAYMENT. MONTH\_NUMBER\_PERIOD. At column year is calculated by INSURANCE\_PAYMENT. MONTH\_NUMBER\_PERIOD / 12 (Get the integer part). In the column month is calculated by the odd part of the calculation above |
|  | lblDaysReturnMax | Label Column |  | NULL | * Label the number of days of maximum benefit * The data in this column mapping corresponds to INSURANCE\_BENEFIT\_CONFIG.DAYS\_MAXIMUM\_PAYMENT |
|  | lblSalaryToPayment | Label Column |  | NULL | * Label of total salary in grid * The data in this column mapping corresponds to INSURANCE\_PAYMENT. BASE\_SALARY\_PAYMENT |
|  | lblRatePercentToReturn | Label Column |  | NULL | * Percentage label enjoyed in grid * The data in this column mapping corresponds to INSURANCE\_PAYMENT. OWNER\_PAYMENT\_PERCENT |
|  | lblNumberMonthOff | Label Column |  | NULL | * Label number of employee's maternity leave * The data in this column mapping corresponds to INSURANCE\_PAYMENT. OWNER\_PAYMENT\_MONTHS |
|  | lblPaymentMoney1 | Label Column |  |  | * Label of employee's one-time benefit amount   The data in this column mapping corresponds to BASE\_SALARY\_MINIMUM.AMOUNT\_MONEY \* 2 \* INSURANCE\_PAYMENT. NUMBER\_CHILDS\_PER\_ONE |
|  | lblPaymentMoneyBirth | Label Column |  | NULL | * Label the employee's maternity leave allowance amount * The data in this column mapping corresponds to INSURANCE\_PAYMENT. OWNER\_PAYMENT\_MONEY |
|  | lblPaymentMoneyTotal | Label Column |  | NULL | * Label the employee's maternity leave allowance amount * The data in this column mapping corresponds to INSURANCE\_PAYMENT. TOTAL\_INSURANCE\_PAYMENT = OWNER\_PAYMENT\_MONEY + BASE\_SALARY\_MINIMUM.AMOUNT\_MONEY \* 2\*INSURANCE\_PAYMENT.NUMBER\_CHILDS\_PER\_ONE |
|  | lblAttachment | Label Column |  | NULL | * Attached document label with employee's regime settlement information * The data in this column is clicked by the user to download. This data is fetched from the corresponding database mapping in the ATTACHMENT table with TYPE = 'GQCDSC' |
|  | cboApproval | Checkbox |  | UNCHECKED | * Check box to mark the record of information about whether the mode is approved or not * The data in this column mapping corresponds to INSURANCE\_PAYMENT.STATUS |
|  | txtNotesApproval | Textbox |  | NULL | * Note when the user browses or doesn't browse the record * The data in this column mapping corresponds to INSURANCE\_PAYMENT. APPROVAL\_NOTES |
|  | btnApproval | Button |  | NULL | * Approve Button * The event handling when this button is clicked is described in detail below |
|  | btnExport | Button |  | NULL | * Button to export report data * The event handling when this button is clicked is described in detail below |
|  | btnSave | Button |  | NULL | * Button saves the information entered by the user to solve the mode |

#### Business flow



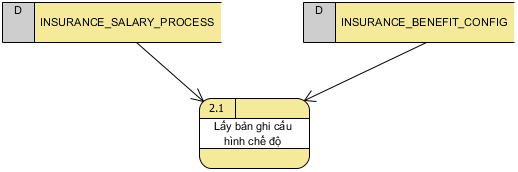
##### Enter the beneficiary's information



1. Prepare data (User starts by clicking on the “Resolve” menu)
   * Prepare data for combobox **cbxModeType** with data loaded from SYS\_CAT table with SYS\_CAT\_TYPE = 'Society type'. Default data is Sick mode
   * Prepare data for combobox **cbxMonthCaculate** with data upload in combobox month 1-12, data upload in combobox year from [Current Year – 5] to [Current Year + 5]
   * Prepare data for the textbox **popupOrgChoose** with the unit data to be uploaded as the decentralized unit associated with the user logged in to the system
   * Data in grid area Resolve childbirth mode blank
   * Data in grid area List of employees Solve childbearing mode blank
   * The data in the employee selection textboxes is blank
2. Select mode type (When user clicks on combobox Mode Type - Childbirth): When data in **popupOrgChoose,** combobox **cbxMonthCaculate** is non-empty, data of grid List of workers Solves childbearing mode load up. And vice versa
   * Conditions to load data at Grid List of employees The settlement of childbirth mode is to retrieve records in the INSURANCE\_PAYMENT table that satisfy the following conditions:
     1. INSURANCE\_PAYMENT. INS\_PAY\_SUMMARY\_ID = INS\_PAY\_SUMMARY. INS\_PAY\_SUMMARY\_ID with INS\_PAY\_SUMMARY.PAYMENT\_MONTH && INS\_PAY\_SUMMARY.PAYMENT\_YEAR = cbxMonthCaculate.value; && INS\_PAY\_SUMMARY.MODE\_TYPE\_ID = 'Paternity leave type';
     2. INSURANCE\_PAYMENT. organization\_id = popupOrgChoose.value;
3. Select unit to solve mode (When user click on option Popup select unit):
   * The unit selection popup is displayed for the user to select the unit
   * When the unit is selected and the Popup is closed, the Grid of the Employee List Settlement of Childbirth is reloaded with the input of the changed unit
4. Select employee to resolve mode (when clicking on Textbox txtEmployeeCode):
   * The system opens Datapicker to search for employees (Use datapicker to find shared employees in the system – including searching by employee code and name)
   * The list of employees displayed in the datapicker is obtained with the following conditions:
     1. Is the number of units that have been selected in the Unit Selection Popup
   * When the user selects an employee from the Datapicker:
     1. The following information is displayed by default, including: Full name, Social insurance book number, Unit, Date of birth
     2. The information block at Grid dealing with childbirth mode is displayed 1 line with information fields according to the screen layout
5. Selecting attachments (When the user clicks on the Browser file button…): borrow files as usual
6. Selection of conditions of enjoyment (When the user clicks and combo-box Enjoy conditions)
   * If the textbox **txtDateOfBirth (date of birth)** has data (non-empty), then:
     1. The data in the column of the number of months of leave is uploaded according to the configuration corresponding to the entitlement conditions
     2. The data in the column up to the date of leave is uploaded =Date of birth + number of months of leave
   * If the textbox **txtDateOfBirth (date of birth)** does notIf there is (empty) data, no event occurs
7. Enter information in the textbox **txtDateOfBirth** - Date of Birth (Onchange event):
   * If the combo-box cbxCondition **(Conditions)** has data (non-empty – selected), then:
     1. The data in the column of the number of months off (According to the calculation in component number 14 – lblNumberMonth) is calculated according to the configuration corresponding to the eligibility conditions.
     2. Data in column up to date of holiday is uploaded =Date of birth + number of months of leave (Retrieved from the corresponding configuration record from the table INSURANCE\_BENEFIT\_CONFIG. MAX\_NON\_WORKING\_DAY)
     3. The data in the Reloaded Object Area column is: showing the employee's current EMP\_TYPE\_PROCESS object process (The process with EMP\_TYPE\_PROCESS. effective\_start\_date < txtDateOfBirth.value < EMP\_TYPE\_PROCESS. effective\_end\_date)

If the combo-box cbxCondition **(Conditions to enjoy)** does notIf there is (empty) data, no event occurs

##### Get mode configuration record



Query the INSURANCE\_BENEFIT\_CONFIG level configuration data table to get 1 configuration record with the following query condition:

* + Number of months of paying social insurance of an employee = ∑ (The number of months calculated according to 2 fields of information INSURANCE\_SALARY\_PROCESS.EFFECTIVE\_START\_DATE and INSURANCE\_SALARY\_PROCESS.EFFECTIVE\_START\_DATE) according to the condition INSURANCE\_SALARY\_PROCESS. EMPLOYEE\_ID = employee\_id of employee selected from Datapicker
    1. The rule for rounding the month is calculated according to the above formula as follows: If the odd month is >=15, add 1 month, and if <15, ignore the 🡪number of months of paying social insurance will be an integer month.
  + INSURANCE\_BENEFIT\_CONFIG. EMP\_TYPE\_ID = EMPLOYEE.EMP\_TYPE\_ID
  + INSURANCE\_BENEFIT\_CONFIG.LABOUR\_CONTRACT\_TYPE\_ID = EMPLOYEE.LABOUR\_CONTRACT\_TYPE\_ID
  + INSURANCE\_BENEFIT\_CONFIG. INSURANCE\_CONDITION\_ID = cbxCondition.value (Selected data at combobox benefit conditions)

INSURANCE\_BENEFIT\_CONFIG. MONTH \_PAYMENT\_FROM <= Number of months to pay social insurance <= INSURANCE\_BENEFIT\_CONFIG.MONTH\_PAYMENT\_TO

##### Calculate the number of days resolved

Use the configuration record taken from step 1 above to perform the calculation of the number of days as follows:

* + If INSURANCE\_BENEFIT\_CONFIG. IS\_CHECK\_SATURDAY = false && IS\_CHECK\_SATURDAY = fasle 🡪Number of days resolved = Last day – first day + 1 - (Total number of Saturdays between first 🡪and last) - (Total number of Sundays between beginning 🡪of the last day) - ∑ (Total number of holidays between the first 🡪day of the last day) (Only subtract the total number of holidays when INSURANCE\_BENEFIT\_CONFIG. IS\_CHECK\_HOLIDAY = false)
    1. Where: Last day = First day + number of months of maternity leave
  + If INSURANCE\_BENEFIT\_CONFIG. IS\_CHECK\_SATURDAY = true && IS\_CHECK\_SUNDAY = true 🡪Number of days resolved = Last day – first day + 1 - ∑ (Total number of holidays between the first 🡪day of the last day) (Only subtract the total number of holidays when INSURANCE\_BENEFIT\_CONFIG. IS\_CHECK\_HOLIDAY = false )
    1. Where: Last day = First day + number of months of maternity leave

##### Split processing time:

This split is to create multiple mode resolution records with different branch conditions to insert into the INSURANCE\_PAYMENT table.

* + Split when changing minimum wage information (LTT) according to the following conditions:
    1. Query in the BASE\_SALARY\_MINIMUM table for minimum wage records that satisfy the condition BASE\_SALARY\_MINIMUM.APPLY\_DATE < lblUptoDate.Value && BASE\_SALARY\_MINIMUM.EXPIRY\_DATE > dateBirthOfChild.value. If the number of query records (n) satisfying the above condition > 1 🡪perform split mode resolution into (n) records with INSURANCE\_PAYMENT information. START\_ACTUAL\_DATE[n] = Birthdate 🡪BASE\_SALARY\_MINIMUM.EXPIRY\_DATE[n]. INSURANCE\_PAYMENT.START\_ACTUAL\_DATE [n-1] = BASE\_SALARY\_MINIMUM.APPLY\_DATE [n-1] && INSURANCE\_PAYMENT.END\_ACTUAL\_DATE[n-1]= BASE\_SALARY\_MINIMUM.EXPIRY\_DATE [n-1],If BASE\_MINIM\_S Otherwise INSURANCE\_PAYMENT.END\_ACTUAL\_DATE[n-1]= lblUptoDate.value ) . Such a split iterates over (n) records

##### Calculation of salary and benefits

* + - * 1. If the configurable salary is the minimum salary, the salary is equal to the minimum salary = BASE\_SALARY\_MINIMUM. AMOUNT\_MONNEY (Minimum wage record query meets the condition First Date< BASE\_SALARY\_MINIMUM.APPLY\_DATE < Last Date + BASE\_SALARY\_MINIMUM. EXPIRE\_DATE = null
        2. If the salary configuration is the insurance salary, the formula for calculating the salary receiving the regime is as follows:

TL entitled to the regime = (∑{LTT x [HSBH x (1+TNVK) + HSCV] x (1+ TNVT)})/Number of months of configuration.

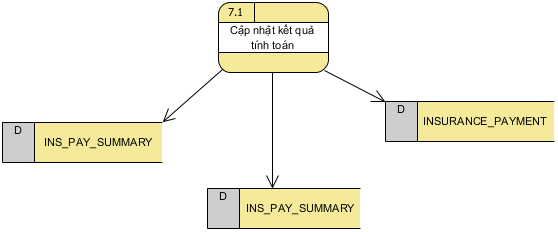
**Inside:**

* TL: The salary used as the basis for payment of social insurance contributions of the month preceding the leave. Calculated by the average number of months of social insurance next to the month of calculation of vacation days according to the number of months configured in the management service configuration level of benefits.
* LTT: Minimum Wage = Query data table BASE\_SALARY\_MINIMUM. AMOUNT\_MONNEY with the condition BASE\_SALARY\_MINIMUM.EXPIRE\_DATE == NULL or BASE\_SALARY\_MINIMUM.EXPIRE\_DATE > cbxMonthCaculate.Value (Time of resolution mode)
* HSBH: The social insurance coefficient of the beneficiary is managed in the management of the social insurance salary process = Query the data table *INSURANCE\_SALARY\_PROCESS* . FACTOR with the condition *INSURANCE\_SALARY\_PROCESS* . EFFECTIVE\_END\_DATE == NULL or *INSURANCE\_SALARY\_PROCESS* . EFFECTIVE\_END\_DATE . cbxMonthCaculate.Value (Time to resolve mode)
* TNVK: Out-of-frame seniority (applicable to QNCN objects) in % = Retrieve data at the queried record for the HSBH above = Query the data table *INSURANCE\_SALARY\_PROCESS* . SENIORITY\_PERCENT with the condition *INSURANCE\_SALARY\_PROCESS* . EFFECTIVE\_END\_DATE == NULL or *INSURANCE\_SALARY\_PROCESS* . EFFECTIVE\_END\_DATE . cbxMonthCaculate.Value (Time to resolve mode)
* HSCV: Position coefficient = Refer to the HSCV calculation function in the calculation of deduction and payment for social insurance
* Occupational income: Seniority (applicable to SQ and QNCN subjects) in % = Professional reference for calculation of deduction and payment for social insurance
* If checking within n months (n= number of months of premium payment before enjoying the regime in the configuration), that person does not have the number of months of premium payment >= the average number of months of salary calculation in the configuration, the message will be displayed. Report "Employees whose number of months of paying social insurance premiums in about n months is not enough to enjoy the regime"

##### Calculation of allowance

1. For the configuration record queried in step 1 for the 🡪number of months of the one-time benefit, the one-time allowance will be calculated as follows:
   * One-time allowance = [number of months of 1-time benefit + (reference number – reference from) /progressive step \* progressive increase]\* salary of one-time allowance salary type (configured) = [ *INSURANCE\_BENEFIT\_CONFIG .MONTH\_ALLOWANCE* + ( *INSURANCE\_BENEFIT\_CONFIG* .REFERENCE\_FROM - *INSURANCE\_BENEFIT\_CONFIG.REFERENCE\_TO* )/ ( *INSURANCE\_BENEFIT\_CONFIG* .PROGRESSANCE\_STEPEN .
   * In which: XYZ is calculated as follows
     1. If INSURANCE\_BENEFIT\_CONFIG *.* \_ *SALARY\_TYPE\_ALLOWANCE = 1 (Minimum Wage Type) 🡪XYZ = BASE\_SALARY\_MINIMUM* .AMOUNT\_MONNEY (Minimum wage record query satisfies the condition First Date< *BASE\_SALARY\_MINIMUM* .APPLY\_DATE < Last Date + *BASE\_SALARY\_RE\_MINIMUM* ) = null )
     2. If INSURANCE\_BENEFIT\_CONFIG *.* \_ *SALARY\_TYPE\_ALLOWANCE = 1 (Minimum wage type) 🡪XYZ = Benefit allowance* (calculated in step 5)
2. Allowance = (number of months + odd days/26) \* % of benefits \* TL for benefits (calculated in step 5). Inside:
   * The number of months, the number of odd days is calculated based on 2 days: Last day - first day
   * %entitled = *INSURANCE\_BENEFIT\_CONFIG.* *RECEIVEMENT\_PERCENT*
3. Total benefit = One-time allowance (calculated in item a) + allowance (calculated in item b)

##### Save the data of the above calculation steps into the database with the corresponding tables as follows:



1. Save general information at the time of mode resolution to the INS\_PAY\_SUMMARY table including the following information:
   * INS\_PAY\_SUMMARY. INS\_PAY\_SUMMARY\_ID = Autoincrement = Next\_seq;
   * INS\_PAY\_SUMMARY. PAYMENT\_MONTH = cbxMonthCaculate.value;
   * INS\_PAY\_SUMMARY . PAYMENT\_YEAR = cbxMonthCaculate.value;
   * INS\_PAY\_SUMMARY. CREATED\_BY = SYS\_USER\_ID;
   * INS\_PAY\_SUMMARY. CREATED\_TIME = Now();
   * INS\_PAY\_SUMMARY. STATUS = 1;
   * INS\_PAY\_SUMMARY. MODE\_TYPE\_ID = mode\_type\_id (taken from form down)
   * INS\_PAY\_SUMMARY. employee\_id = employee\_id (taken from form)
2. Save the calculation mode settlement information to the INSURANCE\_PAYMENT table including the following information:
   * INSURANCE\_PAYMENT.INSURANCE\_CONDITION\_ID = Autoincrement = Next\_seq;
   * INSURANCE\_PAYMENT.START\_ACTUAL\_DATE = Date of Birth = txtDateOfBirth.value;
   * INSURANCE\_PAYMENT. END\_ACTUAL\_DATE = START\_ACTUAL\_DATE + number of months off (Taken from the configuration record retrieved in step 1 Get the 🡪*INSURANCE\_BENEFIT\_CONFIG* information field . *MAX\_NON\_WORKING\_DAY)*
   * INSURANCE\_PAYMENT. START\_WORKINGDAY\_DATE = NULL;
   * INSURANCE\_PAYMENT. END\_WORKINGDAY\_DATE = NULL;
   * INSURANCE\_PAYMENT. NOTE = txtNotesUp.value;
   * INSURANCE\_PAYMENT. CHILD\_DATE\_OF\_BIRTH = dateBirthOfChild.value;
   * INSURANCE\_PAYMENT. CREATED\_BY = SYS\_USER\_ID;
   * INSURANCE\_PAYMENT . NUMBER\_CHILDS\_PER\_ONE = txtNumberBirth.value;
   * INSURANCE\_PAYMENT. CREATED\_TIME = Now();
   * INSURANCE\_PAYMENT. UPDATED\_BY = SYS\_USER\_ID;
   * INSURANCE\_PAYMENT. UPDATED\_TIME = Now();
   * INSURANCE\_PAYMENT. INSURANCE\_ENTERING\_DATE = null;
   * INSURANCE\_PAYMENT. MONTH\_NUMBER\_PERIOD = (Query the table INSURANCE\_SALARY\_PROCESS to get the processes of this employee, then calculate ∑( INSURANCE\_SALARY\_PROCESS. EFFECTIVE\_END\_DATE - INSURANCE\_SALARY\_PROCESS .EFFECTIVE\_START\_DATE )
   * INSURANCE\_PAYMENT. DAYS\_MAXIMUM\_PAYMENT = Last day – first day
   * INSURANCE\_PAYMENT. BASE\_SALARY\_PAYMENT = XYZ (Calculated in step 6)
   * INSURANCE\_PAYMENT. OWNER\_PAYMENT\_PERCENT = *INSURANCE\_BENEFIT\_CONFIG.* *RECEIVEMENT\_PERCENT*
   * INSURANCE\_PAYMENT. OWNER\_PAYMENT\_MONTHS = *INSURANCE\_BENEFIT\_CONFIG.* *MAX\_NON\_WORKING\_DAY*
   * INSURANCE\_PAYMENT. OWNER\_PAYMENT\_MONEY = Allowance (calculated in item b of step 6)
   * INSURANCE\_PAYMENT. TOTAL\_INSURANCE\_PAYMENT = Total benefit amount (calculated in step 6)
   * INSURANCE\_PAYMENT. APPROVAL\_NOTES = null;
   * INSURANCE\_PAYMENT . TOTAL\_DAYS\_PROGRESSIVE = 0;
   * INSURANCE\_PAYMENT. labor\_contract\_type\_id = labor\_contract\_type. labor\_contract\_type\_id (corresponds to the employee's object record at the current time)
   * INSURANCE\_PAYMENT. emp\_type\_id = emp\_type. labor\_contract\_type\_id (corresponds to the employee's object record at the current time)
   * INSURANCE\_PAYMENT. organization\_id = employee.organization\_id
   * INSURANCE\_PAYMENT. INS\_PAY\_SUMMARY\_ID = INS\_PAY\_SUMMARY .INS\_PAY\_SUMMARY\_ID
3. Browse the list (When the user ticks each checkbox on the list and clicks the Browse list button)
   * Update the INSURANCE\_PAYMENT table with the following fields:
     1. INS\_PAY\_SUMMARY. STATUS = 2;
     2. INSURANCE\_PAYMENT. APPROVAL\_NOTES = txtNotesApproval. Value;
     3. Under the condition INS\_PAY\_SUMMARY.employee\_id = employee\_id (checked for approval)
4. Export data (When user clicks and Export button)

# GENERAL DESIGN AND REUSABLE

N/A

# DESIGN TO GUARANTEE DATA MANAGEMENT STANDARDS

[List designs/features/solutions that comply with Data Governance standards stated in CTKT and PTYC]

[Eg:

* In order to comply with the data opacity standard with confidential data being customer data on the BCCS\_CC system:
* At the customer information lookup function, in case the user does not have the right or does not enter the customer identification code, 🡪blurring customer information such as full name, phone number ... (eg: sdt displayed as: \* \*\*\*\*\*\*123).
* ]