Requirements Engineering

RollPal

Submitted By: David O’Sullivan (T00139303)

Computing with Games Development

Date Submitted: 02/05/2018

**Table of Contents**

[1. Introduction/overview 4](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288086)

[2. Functional Components 5](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288087)

[3. User Requirements 6](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288088)

[4. System Requirements 7](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288089)

[4.1. System Level Use Case Diagram 8](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288090)

[4.2. Manage Staff 9](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288091)

[4.2.1. Register Staff 9](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288092)

[4.2.2. Update Staff 11](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288093)

[4.2.3. De-Register Staff 13](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288094)

[4.2.4. Search Staff 15](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288095)

[4.3 Manage Payroll 17](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288096)

[4.3.1 Enter Time Sheet 17](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288097)

[4.3.2 Generate Payroll 19](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288098)

[4.3.3 Pay Employees 21](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288099)

[4.3.4 Search Pay Roll 22](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288100)

[4.4 Taxation Clauses 24](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288101)

[4.4.1 Create Taxation Clause 24](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288102)

[4.4.2 Update Taxation Clause 29](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288103)

[4.4.3 Delete Taxation Clause 31](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288104)

[4.5 Admin 32](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288105)

[4.5.1 Print Document 32](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288106)

[4.5.2 Analyse Payroll 34](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288107)

[4.5.3 Banking details 36](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288108)

[5. System Model 37](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288109)

[5.1. Level-0 DFD 37](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288110)

[5.2. Level-1 DFD 38](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288111)

[5.3. Level-2 DFD (P1 Manage Staff) 39](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288112)

[5.4. Level-2 DFD (P2 Manage Payroll) 40](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288113)

[5.5. Level-2 DFD (P3 Taxation Clauses) 41](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288114)

[5.6. Level-2 DFD (P4 Admin) 42](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288115)

[6. Data Model (Class Diagram) 42](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288116)

[6.1. Class Diagram 43](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288117)

[6.2. Relational Schema 43](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288118)

[6.3. Database Schema 43](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288119)

[7. Conclusion 44](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288120)

[8. Appendices 45](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288121)

[8.1. Appendix A – Title 45](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288122)

[8.2. Appendix B – Title 45](file:///C:\Users\HP\Documents\GitHub\RollPal\GD_RE_O'Sullivan_David.docx#_Toc498288123)

# Introduction/overview

RollPal is a user friendly system that gives enterprise a fast and efficient way of managing and implementing payroll features. RollPal is an efficient and streamlined way of entering staff profiles and processing wages. It will also allow administrative features like the processing of weekly, monthly and annual wage tallies and comparisons as well as the creation of Taxation Clauses such as new or updated taxes that will be implemented automatically in your weekly wage calculations.

RollPal’s primary function is to make the pay roll an easier and less arduous experience for administrative staff. In turn, that will free up the staff to give them more time to spend on other areas of the business.

RollPal’s secondary function is to provide a search function that will allow administrative staff to perform a more efficient and accurate report on expenditure on the pay roll. RollPal will also allow the Administrator to enter the companies banking details

# Functional Components

# User Requirements

1. RollPal will perform Staff Administration.
   1. RollPal will register new Staff
   2. RollPal will allow a staff members details to be updated
   3. RollPal will allow a staff member to be deregistered
   4. RollPal will perform a staff enquiry
2. RollPal will perform Pay Roll Management
   1. RollPal will provide a timesheet entry
   2. RollPal will generate the pay roll
   3. RollPal will process wages
   4. RollPal will perform a payroll enquiry
3. RollPal will perform Taxation Clauses automatically
   1. RollPal will allow the user to create Taxation Clauses
   2. RollPal will allow the user to update Taxation Clauses
   3. RollPal will allow the user to remove Taxation Clauses
4. RollPal will perform administrative analysis
   1. RollPal will print business documents
   2. RollPal will analyse staff details

# System Requirements

|  |  |  |
| --- | --- | --- |
| **Functional Requirements** | **Non-Functional Requirements** | **Domain Requirements** |
| 1. To log all payments and keep track of all different rates of pay i.e. travel, overtime, Sunday. 2. To automatically calculate tax based on Taxation Clauses. 3. To give the user the option of updating or adding or removing Taxation Clauses for changes in tax/u.s.c etc. 4. To store all inputs and statistics on a database 5. The system will export a .pdf which the system will then send to the employee | 1. Speed is key. There will be minimal windows. Each entry will happen on one form. 2. Security is of the utmost importance. Each staff member who has been trained will have their own unique password that will be randomly generated and can be re-assigned if they fear the password has been compromised. 3. Design will be minimalistic .RollPal will be as aesthetically pleasing as it is functional. | 1. Due to security issues the software will not have online capability however the employer will have access at home. The system will only be available in the office network. |

## System Level Use Case Diagram

RollPal will require any relevant details from an employee. This will allow the employee to be registered on the system. The administrator will then have the ability to update these details or to de-register an employee.

RollPal will then allow the administrator to enter time sheets that RollPal will then calculate. The administrator will then make the total wage bill available for inspection by the employer/manager which he will then approve/disapprove.

RollPal will allow the creation, update or deletion of Taxation Clauses which the administrator can enter into the system to implement into its wage calculations. These rules can be imposed by the business, government or unions.

RollPal will allow the administrator to query the database or calculate and print annual documents such as p60’s or p45’s. This function will also allow the administrator to update the company’s banking details.

RollPal

Employer/ Manager

Administrator

Employee

Employer

Administrator

Employee

Employer

Administrator

Employee

Employer

Administrator

## Manage Staff

This function allows the creation of a staff member’s details on the system.

### Register Staff

This allows the admin to register an employee on the system.

**Employee**

Details

**Administrator**

Details

«includes»

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Register Staff** | |
| **Use Case Id** | 4.2.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Staff | |
| **Description** | This function registers a new employee on the system | |
| **Preconditions** | The employee will sign a form with their details and admin will enter it to the system | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The admin will invoke the Register Staff function  **Step 4:** The admin will then enter the new employees details   * Forename * Surname * Phone Number * Address * Eircode * Email * IBAN * Date of Birth * Gender * Marital Status * Children * Street * Town * County * Pay Grade | **Step 2:**The system determines the next staff id  **Step 3:** Display the U.I.  **Step 5:** The system validates the employees details   * All Fields must be entered * Emails must be valid * Eircode must be valid * Phone number must be all digits * Phone number must be ten digits * IBAN must begin with ‘IE’ and be 22 characters in length   **Step 6:** Set Active to ‘y’  **Step 7:** The system decides what PAYE grade the Staff member is  **Step 8:** The system will then take the data and store it in the Staff file:   * StaffID * Forename * Surname * Date of Birth * Gender * Marital status * Children * Active * Pay Grade   **Step 9**: The System will then store the following information in the Contact File:   * StaffID * Email * Phone * Street * Town * County * Eircode   **Step 10:** The System will then store the following information in the Banking file:   * StaffID * IBAN   **Step 11:** The system then creates an entry in the Staff Paid File. The following information is stored   * StaffId * Concurrent Gross * Concurrent Net * Week 1-52   Concurrent gross, Concurrent net and each week from 1 – 52 are set to 0.00  **Step 12:** The System creates an entry in the Taxation File  And stores :   * StaffId * PayeGrade * PrsiGrade * SiptuGrade * UscGrade * HourlyRate   **Step 13:** The system creates an entry for the Concurrent\_Taxation File and stores   * StaffId * ConcurrentUsc * Usc * ConcurrentSiptu * Siptu * ConcurrentPaye * Paye * ConcurrentPrsi * Prsi   ConcurrentUsc, Usc, ConcurrentSiptu, Siptu, ConcurrentPaye, Paye, ConcurrentPrsi, Prsi are all set to 0.00  **Step 14:** Display Confirmation Message.  **Step 15**: Clear the U.I.  **Step 16:** Return Home |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| The admin enter an invalid detail e.g. (IBAN only 21 characters) | **Step 1:** The admin will invoke the Register Staff function  **Step 4:** The admin will then enter the new employees details   * Forename * Surname * Phone Number * Address * Eircode * Email * Iban * Date of Birth * Gender * Marital Status * Children * Street * Town * County * Pay Grade | **Step 2:** The System determines the next StaffID  **Step 3:** The System displays the U.I  **Step 5:** The System validates the employees details   * All Fields must be entered * Emails must be valid * Eircode must be valid * Phone number must be all digits * Phone number must be ten digits * IBAN must begin with ‘IE’ and be 22 characters in length   **Step 6:** Display Error Message  **Step 7:** Display warning Icon next to the IBAN text field |
| **Conclusions** | The staff member is now registered on the system | |
| **Post conditions** | The staff member can be contacted, taxed and paid | |
| **Business Rules** | The employee must be between the age of 16 and 66 | |
| **Implementation Constraints** |  | |

### Update Staff

**Employee**

Details

**Administrator**

Details

«includes»

«extends»

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Staff** | |
| **Use Case Id** | 4.2.2 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Staff | |
| **Description** | This function updates an existing employees information on the system | |
| **Preconditions** | The employee must be registered on the system | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
| **Search by I.D.** | **Step 1:** The admin will invoke the Update Staff function  **Step 4:** The user will then search by StaffId  **Step 6:** The user will select the desired employee from the search results  **Step 8:** The user will then alter any one of the following details   * Forename * Surname * Date Of Birth * Gender * Marital Status * Pay Grade * Marital Status * PhoneNumber * EirCode * Email * Children * Street * Town * County * IBAN   **Step 11:** The user clicks the confirm button | **Step 2:** Display the U.I  **Step 3:** The system will prompt the user with a drop down menu with all registered staff members Staff Id’s  **Step 5:** The system will display the first registered staff member  **Step 7:** The system displays the information of the desired staff member in the appropriate text fields  **Step 9:** The system validates the employees details   * All Fields must be entered * Emails must be valid * Eir-code must be valid * Phone number must be all digits * IBAN must start with ‘IE’ * IBAN must be 22 characters long   **Step 10:** Display confirmation message confirming the changes  **Step 12:** The System will then store the Following information in the Staff File   * Forename * Surname * Date of Birth * Gender * Marital status * Children * Pay Grade   **Step 11**: Store the following information in the Banking File   * IBAN   **Step 12:** Store the following information in the Contact File   * Email * Phone * Street * Town * County * EirCode   **Step 13:** The system return to the home Screen |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The staff members information is now updated on the system | |
| **Post conditions** | The staff members details can be kept up to date | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### De-Register Staff

**Administrator**

De-activate

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **De-Register Staff** | |
| **Use Case Id** | 4.2.3 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** |  | |
| **Description** | This function sets an existing employees status to ‘inactive’ | |
| **Preconditions** | The employee must be registered on the system | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The admin will invoke the De-Register Staff function  **Step 5:** The user will then select a staff I.D number  **Step 6**: The administrator will then select the ‘De-Register Button’  **Step 8:** The user will then click the ‘OK’ Button confirming their selection | **Step 2:** Display the U.I  **Step 3:** The system will prompt the user with a drop down menu with all available staff members  **Step 4:** The system will display the details of the first registered staff member  **Step 6:** The system will then display the details of the selected staff member in Text fields  **Step 7:** The system will then prompt the user with a warning advising them of the dangers of accidentally de-registering an employee  **Step 9:** Display an appropriate confirmation message  **Step 10:** The system will then set the staff members active status to ‘n’  **Step 11:** Clear the U.I  **Step 12:** The system returns to the welcome screen |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **The staff member is already de-activated** | **Step 1:** The admin will invoke the De-Register Staff function  **Step 5:** The user will then select a staff I.D number  **Step 6**: The administrator will then select the ‘Re-Register Button’  **Step 8:** The user will then click the ‘OK’ Button confirming their selection | **Step 2:** Display the U.I  **Step 3:** The system will prompt the user with a drop down menu with all available staff members  **Step 4:** The system will display the details of the first registered staff member  **Step 6:** The system will then display the details of the selected staff member in Text fields  **Step 7:** The system will then prompt the user with a warning advising them of the dangers of accidentally re-registering an employee  **Step 9:** Display an appropriate confirmation message  **Step 10:** The system will then set the staff members active status to ‘y’  **Step 11:** Clear the U.I  **Step 12:** The system returns to the welcome screen |
| **Conclusions** | The staff member is now re-activated on the system | |
| **Post conditions** | The staff member will be re-subscribed from any emails, can be paid and will now receive notifications from the company. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### 4.2.4. Search Staff

**Administrator**

Query

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Search Staff** | |
| **Use Case Id** | 4.2.4 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the administrator to search the weekly payroll to see what any particular staff earned and compare it with others | |
| **Preconditions** | The employee must be registered on the system | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The admin will invoke the Search Staff Function  **Step 4:** The user will then enter a staff I.D number  **Step 6:** The administrator exits the search | **Step 2:** Display the U.I  **Step 3:** The system will prompt the user with a drop down menu with all available staff  **Step 5:** The system will display the details of the desired staff member  **Step 11:** Clear the U.I and return home |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The administrator can view the staff members details without worrying about changing details | |
| **Post conditions** | The administrator can view each staff member. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Manage Payroll

### Enter Time Sheet

**Employee**

Details

**Administrator**

Details

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Enter Time-Sheet** | |
| **Use Case Id** | 4.3.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Employee | |
| **Description** | This function allows the administrator to enter the times worked by each employee throughout the week | |
| **Preconditions** | The employee must be registered on the system | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The admin will invoke the Enter time-sheet function  **Step 4:** The user will select a staff member from the drop down menu  **Step 6:** The administrator will enter the times worked by the staff member  **Step 7:** the administrator will click the ‘Process Wages’ button  **Step 10:** The administrator then confirms this with an ‘Accept’ button  **Step 25:** The user clicks the ‘OK’ Button | **Step 2:** Display the U.I  **Step 3:** The system will provide a drop down menu of all the staff that are active.  **Step 5:** The system will display drop down menus for the start time, finish time, travel time and time spent on break for each day of the week. The user will be able to select what days were worked and whether or not the staff member drove or not that day  **Step 8:** The system will load the staff members pay grade from the Staff Table.  **Step 9:** The system will then obtain the value of the employees pay grade from the Rate Table.  **Step 10:** The system will then retrieve   * Overtime * Sunday * DriverTravel * DriverPassenger   From the Pay\_Multipliers Table  **Step 11:** The system will then calculate the gross weekly wage of the employee  **Step 10:** The system will then retrieve :   * Paye Grade * Prsi Grade * Siptu Grade * Usc Grade   From the taxation table  **Step 11:** The system will then retrieve :   * Value   From the Siptu Table  **Step 13:** The system will then calculate the value of the Siptu transaction and subtract it from the gross wage.  **Step14:** The system will retrieve   * ConcurrentSiptu   From the Concurrent\_Taxation Table  **Step 15:**The System will then add this week’s Siptu to the total paid for the year and store   * ConcurrentSiptu * Siptu   In the Concurrent\_Taxation File  **Step 16:** The system will retrieve:   * UscGrade * Percentage * LowerLimit * UpperLimit   From the USC Table  **Step 17:** The system will then calculate the value of USC transaction and subtract it from the gross wage.  **Step18:** The system will retrieve   * ConcurrentUsc   From the Concurrent\_Taxation Table  **Step 19:**The System will then add this week’s Usc to the total paid for the year and store   * ConcurrentUsc * Usc   In the Concurrent\_Taxation File  **Step 20:** The system will retrieve   * Percentage   From the PRSI Table.  **Step 21:** The system will then calculate the value of the Prsi transaction and subtract it from the gross wage.  **Step22:** The system will retrieve   * ConcurrentPrsi   From the Concurrent\_Taxation Table  **Step 23:**The System will then add this week’s Prsi to the total paid for the year and store   * ConcurrentPrsi * Prsi   In the Concurrent\_Taxation File  **Step 24:** The system will retrieve   * Cutoff * LowerPercentage * UpperPercentage   From the Paye Table.  **Step 25:** The system will then calculate the value of the Paye transaction and subtract it from the gross wage.  **Step22:** The system will retrieve   * ConcurrentPaye   From the Concurrent\_Taxation Table  **Step 23:**The System will then add this week’s Paye to the total paid for the year and store   * ConcurrentPaye * Paye   In the Concurrent\_Taxation File  **Step 24:** The system returns the net wage and gross wage to the user and displays an information dialog detailing these costs.  **Step 26:** The system retrieves   * Concurrent Gross * Concurrent Net   From the Staff\_Paid File  **Step 27:** The system adds the total gross for the year and the net for the year and the net for the week in the Staff\_Paid File.  **Step 28:** The system removes the staff member from the to do list  **Step 29:**  The system resets the U.I |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The employee’s wages have been stored. | |
| **Post conditions** | The employee’s wages can be generated and the employee can be paid. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Generate Payroll

**Administrator**

Request Approval

**Employer / Manager**

Provide Approval

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Generate Pay Roll** | |
| **Use Case Id** | 4.3.2 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Employer / Manager | |
| **Description** | This function allows the administrator to generate a wage total based on the times entered | |
| **Preconditions** | The administrator must have entered all employees times for that week | |
| **Trigger** | The administrator will have the option of navigating here after completing all entries in the ‘Enter Time-sheet’ function. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The admin will invoke the Generate Pay Roll function  **Step 3:** The administrator must click a ‘Generate Payroll’ Button  **Step 5:** The user clicks the ‘OK’ Button. | **Step 2:** Display the U.I  **Step 4:** The System will display the total net cost of the wages to the user  **Step 6:** The system generates an email that is sent to the Manager that has the authority to confirm wage expenditure  **Step 7:** Clear the U.I. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | Approval has been sought from an employer / manager | |
| **Post conditions** |  | |
| **Business Rules** | The Employer / manager must approve the wage payment | |
| **Implementation Constraints** |  | |

### Pay Employees

**Administrator**

Approve

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Pay Employees** | |
| **Use Case Id** | 4.3.3 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Employee | |
| **Description** | This pays the employee and processes pay – slips. | |
| **Preconditions** | The employee’s wages must have been entered and approved | |
| **Trigger** | Generate Pay Roll | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The Administrator invokes the Pay Employee’s function.  **Step 4**: The Administrator clicks the Pay Employee’s Button.      **Step 8:** The administrator clicks the ‘ok’ button | **Step 2:** The System displays the U.I. and then reads the wage total from the Timesheet file.  **Step 3:** A Pay Employee’s Button will appear on screen that asks the user to confirm the processing of the wages.  **Step 5:** The system retrieves the staff members IBAN from the Contact File.  **Step 6:**  The system retrieves the net for the week from the Staff\_Paid File.  **Step 7:** Thesystem displays a message with the IBANs and total money to be transferred for each individual staff member  **Step 9:** The system opens the web page of the company’s bank where the user can login to transfer wages  **Step 10:** The system retrieves the staff members   * Forename * Surname   From the Staff File  **Step 11:** The system retrieves   * Email   From the Contact File  **Step 12:** A Payslip is generated and sent and sent by email with the amount that has been transferred  **Step 8:** Clear the U.I.  **Step 10:** The system return home. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The employees have been paid | |
| **Post conditions** | The employee’s wages have been processed and payslips have been generated and sent to the employee’s | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Search Pay Roll

This process allows the user to view the generated payroll by staff I.D / name and compare to others. This will allow for comparison for example if two people are on the same job and one is earning less because he is working less. This will help analyse whether staff are pulling their weight or examine if unnecessary overtime hours are being recorded for a job.

**Administrator**

Query

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Search Staff** | |
| **Use Case Id** | 4.3.4 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the administrator to search the weekly payroll to see what any particular staff earned and compare it with others | |
| **Preconditions** | The employee must be registered on the system | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The admin will invoke the Search Pay Roll Function  **Step 4:** The user will then enter a staff name or I.D number  **Step 6:** The administrator will select a staff member  **Step 8:** The administrator exits the search | **Step 2:** Display the U.I  **Step 3:** The system will prompt the user with a search function  **Step 5:** The System will search for and display a staff members details if they are found in the database  **Step 7:** The System will display that staff members details for that week  **Step 9:** Clear the U.I and return home |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The administrator can compare staff totals | |
| **Post conditions** | The administrator can view each staff member in order to compare and report on staff doing abnormal hours | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Taxation Clauses

### Create Taxation Clause

**Employer / Manager**

Details

**Administrator**

Details

«includes»

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Create** Taxation Clause | |
| **Use Case Id** | 4.4.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Employer / Manager | |
| **Description** | This function registers a new Taxation Clause on the system | |
| **Preconditions** | Laws or company policy dictate these rules that are applied to the weekly wage bill | |
| **Trigger** | A change in government policy or company policy | |
| **Expected Scenario** | **Actor Action** | **System Response** |
| **Create a** Taxation Clause **that adds 1.5% to the employees wage – this rule shall be called “weekly stipend” for the purposes of this example** | **Step 1:** The administrator invokes the Create Taxation Clause Function  **Step 3:** The administrator chooses adding a Taxation Clause  **Step 5:** The administrator adds the name of the Taxation Clause  **Step 8:** The administrator accepts.  **Step 10:** The administrator chooses between adding a fixed sum or percentage. (Percentage)  **Step 12:** The administrator adds the number percentage  **Step 15:** The administrator chooses between three options : Total wage, Fixed sum or Percentage of total sum (Total wage)  **Step 18 :** The administrator confirms | **Step 2:** Display the U.I  **Step 4:** The System resets The U.I. and displays a textbox labelled ‘Taxation Clause Name’  **Step 6:** The text entered is validated. The Taxation Clause name   * Must contain text * Can contain numbers * Cannot be all numbers   **Step 7:** The system displays a confirmation message  **Step 9:** The system resets the U.I., displaying the previously entered details.  **Step 11:** The system resets the U.I., displaying the previously entered details.  **Step 13**: The System validates the percentage. The percentage   * Must be numeric * Can contain a decimal point   **Step 14:** The system resets the U.I., displaying the previously entered details.  **Step 16:** The system resets the U.I., displaying the previously entered details.  **Step 17:** The System asks the administrator to confirm  **Step 19:** The system resets the U.I |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The Taxation Clause is now registered on the system | |
| **Post conditions** | The Taxation Clause can be now implemented in the weekly wage | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

|  |  |
| --- | --- |
| Administrator | System |
| Invoke Create Taxation Clause Function  No  Choose between fixed sum or percentage, and then input the details  Choose between what the percentage is of: Fixed sum, total wage or percentage  The Administrator enters the name of the Taxation Clause | Yes  Confirm details?  Confirmation Message  Valid?  Confirm and display the U.I  Display the User Interface  Validate the Input  No  Valid?  Yes  Validate Name |
|  | Reset the User Interface |

Taxation Clauses:

Standard Taxation Clauses that apply to all Employees:

1. PAYE

There are three types of PAYE bands

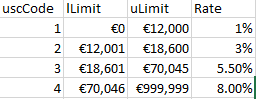
1. Single – 20% of €33,800 and 40% thereafter
2. Single with children - 20% of €37,800 and 40% there after
3. Married – 20% of €42,800 and 40% thereafter
4. PRSI

4% of Gross

1. SIPTU

€13.77 Weekly

1. USC



Along with this there are 3 pay tiers of employees

Tier 1 - €9.55 per hour

Tier 2 – €11.25 per hour

Tier 3 – €14.00 per hour

Each employee earns their base rate (BR) per hour



Each permanent employee is entitled to 25 (8 hour) days holidays

### Update Taxation Clause

Details

**Employer / Manager**

Details

**Administrator**

«includes»

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update** Taxation Clauses | |
| **Use Case Id** | 4.4.2 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Employer / Manager | |
| **Description** | This function Updates a currently registered Taxation Clauses on the system | |
| **Preconditions** | A change in government or company policy can require a Taxation Clause to be updated. | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
| **Update the “USC” percentage** | **Step 1:** The administrator invokes the Update Taxation Clause Function  **Step 3:** The administrator chooses the “usc” Taxation Clause  **Step 5:** The administrator then adjusts the grade ‘b’ percentage to 2.5%  **Step 8:** The administrator accepts. | **Step 2:** Display the U.I and prompt the user with a drop down menu listing the available Taxation Rules  **Step 4:** The system retrieves   * Usc Grade * Percentage * LowerLimit * UpperLimit   From the Usc File and displays it in text boxes on the U.I.  **Step 6:** The system validates the input   * The input must be numeric * The input can contain one decimal point   **Step 7:** The system asks the user to confirm  **Step 9:** The system writes   * Usc Grade * Percentage * LowerLimit * UpperLimit   To the USC File  **Step 10:** The system resets the U.I |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The altered Taxation Clauses is now on the system | |
| **Post conditions** | The altered Taxation Clauses can now be implemented by wage generation | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Delete Taxation Clause

**Administrator**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Delete** Taxation Clauses | |
| **Use Case Id** | 4.4.3 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Employer / Manager | |
| **Description** | This function Deletes a currently registered Taxation Clause on the system | |
| **Preconditions** | The Employer / Manager must have requested the removal of a Taxation Clause | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
| **Remove the “weekly stipend”** Taxation Clause **that adds 2.5% to the employees wage** | **Step 1:** The administrator invokes the Deletes Taxation Clause Function  **Step 3:** The administrator chooses the “weekly stipend” Taxation Clause  **Step 5:** The administrator then unclicks the active button  **Step 9:** The user confirms | **Step 2:** Display the U.I and list Taxation Clauses  **Step 4:** The system displays the details of the selected rule with an active checkbox  **Step 8:** The system asks the user to confirm  **Step 10:** The system resets the U.I. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The Taxation Clause is de-activated on the system | |
| **Post conditions** | The Taxation Clause now cannot be implemented in weekly wage calculations | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Admin

### Print Document

**Administrator**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Print Document** | |
| **Use Case Id** | 4.5.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Employee | |
| **Description** | This function calculates and prints various wage related documents | |
| **Preconditions** | Periodical documents will be generated and be made available to print automatically | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The administrator invokes the Print Document Function  **Step 3:** The administrator chooses between p60, p45 and must also select, through a search function what employee they wish to calculate for.  **Step 5:** The administrator confirms | **Step 2:** Display the U.I and list available documents  **Step 4:** The system asks the user to confirm  **Step 6:** the system then calculates the desired totals and exports them as a .pdf file  **Step 7:** The system resets the U.I. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The desired document is now available | |
| **Post conditions** | The document can be printed or distributed electronically | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Banking details

**Administrator**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Banking Details** | |
| **Use Case Id** | 4.5.3 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the user to store the companies banking details along with each staff member | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The administrator invokes the Banking Details Function  **Step 3:** The administrator searches by I.D or name for an employee  **Step 5:** The user exits the function | **Step 2:** Display the U.I  **Step 4:** The system displays the bank details of the staff member.  **Step 6:** The system resets the U.I. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The Administrator can see the analyse the staff through earnings | |
| **Post conditions** | Comparisons in staff can be useful for seeing which jobs tend to take the longest and seeing if any staff are in common with these jobs | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

RollPal

Employee

Employee details

Send Payslips

## Level-1 DFD

Staff Info

Approve taxation

Get Email

Read Tax

Write tax

Staff Info

Staff Info

P1

Manage Staff

P3

Manage Taxation

D1

Staff File

D3

Contact

D5

Taxation File

D5

Taxation File

Staff info

Get Email

Employer / Manager

Employee

D4

Staff\_Paid File

D2

Banking File

­­­

D2

Banking File

P4

Administration

P2

Manage Payroll

Read Info

Send Payslip

Write times

Read Times

View Banking

View Banking

D4

2

Staff\_Paid

Read Times

Employee

Send P60

Read Tax

D1

Staff File

D6

Staff\_Paid File

## Level-2 DFD (P1 Manage Staff)

De- register

Write to file

P1.1

Register Staff

D1

Staff File

D2

Banking File

D3

Contact File

D4

Staff\_Paid File

P1.3

De-Register Staff

D5

Taxation File

D12

Concurrent\_Taxation File

­­­­

Read from file

Staff Info

Write to file

P1.2

Update Staff

D1

Staff File

Employee

P1.4

Search Staff

­­­­

## Level-2 DFD (P2 Manage Payroll)

## Level-2 DFD (P3 Taxation Clauses)

## Level-2 DFD (P4 Admin)

# Data Model (Class Diagram)

Brief introduction……

## Class Diagram

## Relational Schema

STAFF { StaffID, Forename, Surname, Age, Gender, MaritalStatus, Children }

CONTACT { StaffID, Phone, Email, Street, Town, County, EirCode }

BANKING { StaffID, AccountNo, SortCode }

TIMESHEET { StaffID, TaxID, Overtime, Normal, Saturday, Sunday, Holiday }

TAXATION { TaxID, USCCode, LowerLimit, UpperLimit , SiptuGrade }

## Database Schema

Relation : STAFF

Attributes:

StaffID numeric (3) [Primary Key] UNIQUE, NOT NULL

Forename char (20) NOT NULL

Surname char (20) NOT NULL

Age numeric (2) NOT NULL

Gender char (1) NOT NULL

MaritalStatus (1) NOT NULL

Children numeric (2) NOT NULL

Relation: CONTACT

Attributes:

StaffID numeric (3) [Primary Key] UNIQUE, NOT NULL

Email varchar (30) NOT NULL

Phone varchar (10)

Street char (20) NOT NULL

Town char (20) NOT NULL

County char (10)

EirCode varchar (7)

Relation: BANKING

Attributes:

StaffID numeric (3) [Primary Key] UNIQUE, NOT NULL

AccountNO numeric (10) UNIQUE, NOT NULL

SortCode numeric (10) UNIQUE, NOT NULL

Relation: TIMESHEET

Attributes:

StaffID numeric (3) [Foreign Key] UNIQUE, NOT NULL

TaxID numeric (3) [Foreign Key] UNIQUE, NOT NULL

Overtime numeric (3)

Normal numeric (3)

Saturday numeric (2)

Sunday numeric (2)

Holiday numeric (3)

Relation: TAXATION

Attributes

TaxID numeric (3) [Primary Key] UNIQUE NOT NULL

USCCode char (1)

LowerLimit varchar (7)

UpperLimit varchar (7)

SiptuGrade char (1)

# Conclusion

# Appendices

## Appendix A – Title

## Appendix B – Title

Might include:

* **Lookup / Reference tables**
* **Sample reports / Listings**