***Southwest***

***Software***

***Development***

***Team***

**Second Year Computing Project**

**2011/2012**

**Fully Automated Payroll System**

**Design & Development Team:**

**James Madden: X00086390:**

**Project Manager, Customer Liaison, Software and Database Development, System Testing & Quality Control.**

**Shane Murphy: X00085315:**

**System Design, Software and Database Development, System Testing & Quality Control.**

**Maciej Macierzynski: X00086366:**

**System Design, Software and Database Development, System Testing & Quality Control.**

**2nd deliverables due 27/2/2012 - Completed**

**Table of Contents**

[1 Introduction 3](#_Toc317705310)

[2 Inputs 4](#_Toc317705311)

[2.1 Forms, 4](#_Toc317705312)

[2.2 Screens 4](#_Toc317705313)

[3 Outputs 7](#_Toc317705314)

[3.1 Forms 7](#_Toc317705315)

[3.2 Screens 7](#_Toc317705316)

[4 Database Design 8](#_Toc317705317)

[4.1 ERD Diagram 9](#_Toc317705318)

[4.2 Database Tables (Data Types & Keys) 9](#_Toc317705319)

[5 Program Design 10](#_Toc317705320)

[5.1 Structured English/Pseudo Code 10](#_Toc317705321)

[6 Validation Control 13](#_Toc317705322)

[7 Test Design 15](#_Toc317705323)

[7.1 TEST CONTROL SHEET 15](#_Toc317705324)

# Introduction

The purpose of this document is to detail the design criteria for the input screens, output screens, database design, structured English/Pseudo code of the function of the system, system validation, testing and the user manual, both hard copy and online version. The document should provide a clear and concise view of the proposed system, validation and testing process.

# Inputs

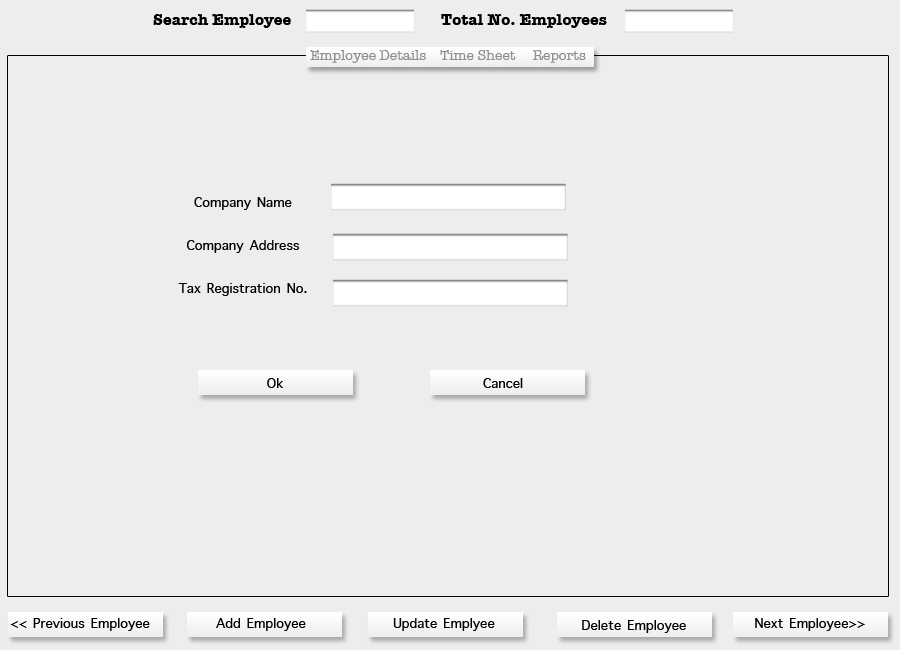
## Forms,

The input forms allow the user to input data to the system such as company details, user login details, employee personnel and revenue details and employee timesheet details

## Screens

The forms above will be displayed as screens to the user, prompting the user to enter the appropriate data. Below we have supplied proposed design screenshots produced with photo shop, there may be minor alterations effected when coding these screens.

* Company Setup Screen



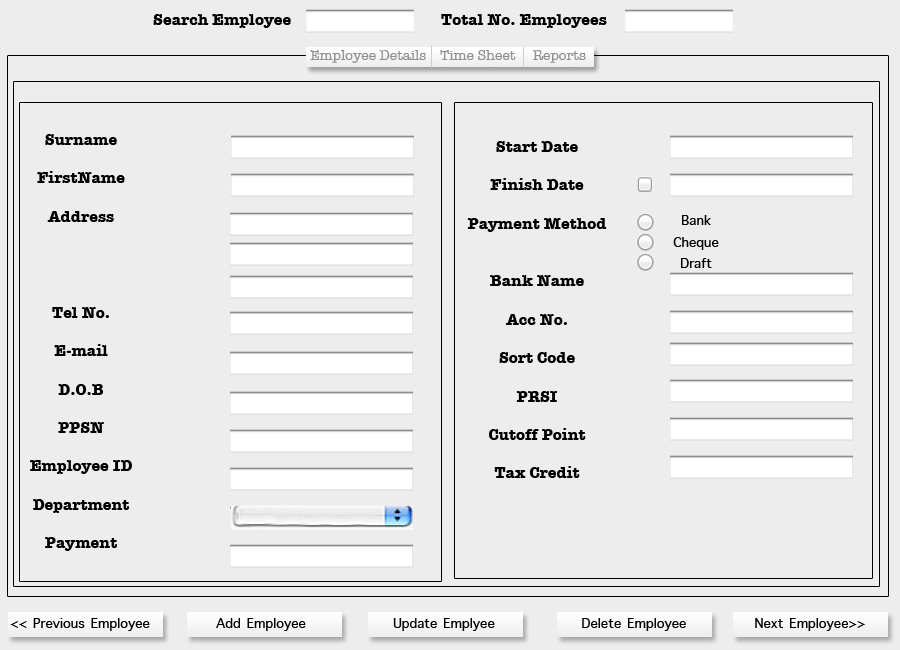
**Right align text on labels**

**Top navigation menu is greyed out?**

* Login Screen



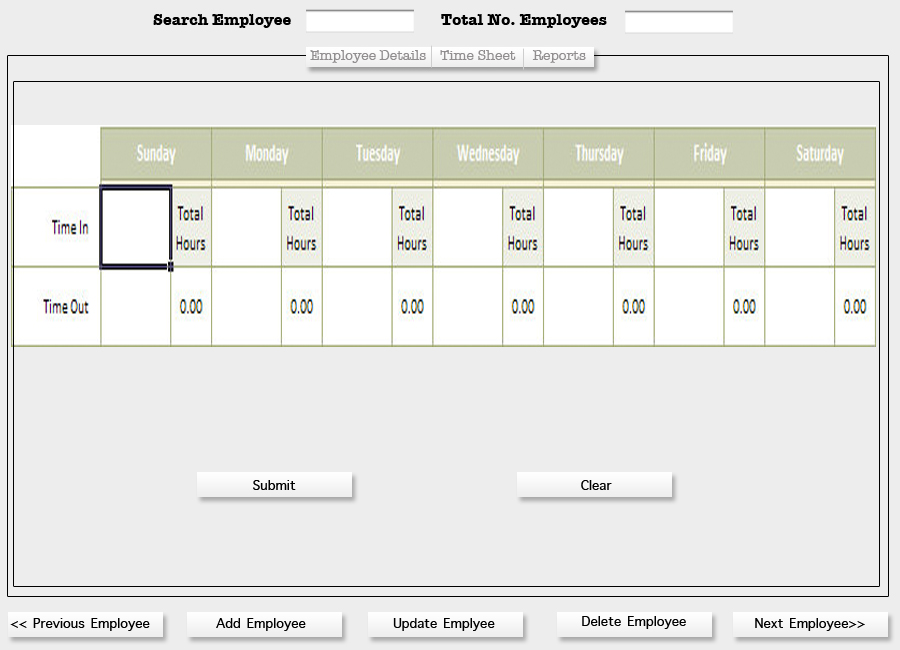
* Add Employee Screen



* Update Employee Screen

“Screen Shot Here”

* Timesheet Screen



# Outputs

## Forms

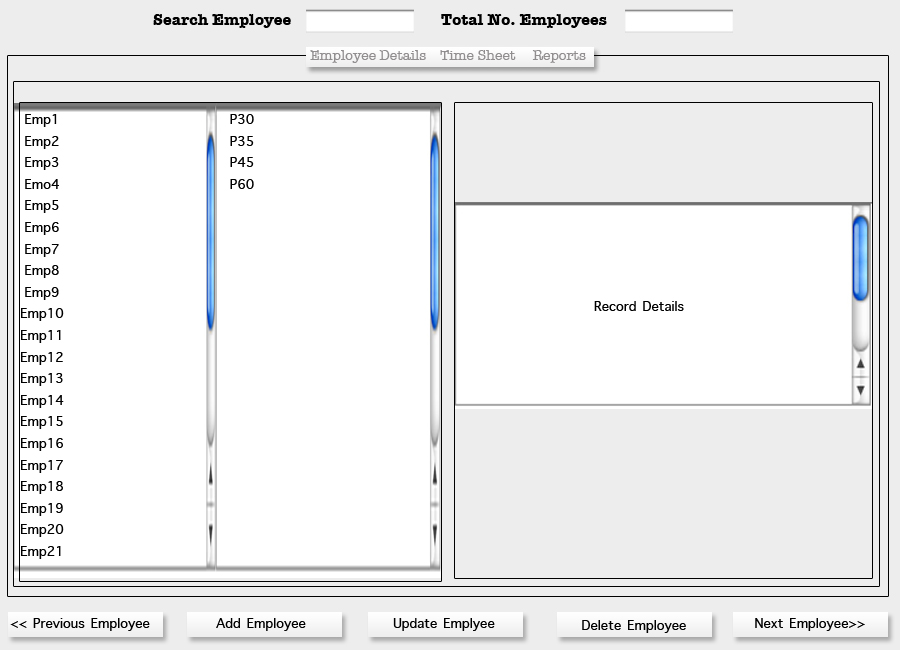
The output forms allow the user to view and extract data from the system such as payment details, weekly payslips, weekly payroll report (total weekly cost), monthly revenue returns (Form P30), annual revenue returns (Form P35), annual employee earnings and deductions (Form P60) and employment cessation (Form P45).

## Screens

## 

The forms above will be displayed as screens to the user, supplying the user with the required relevant data. Below we have supplied proposed design screenshots produced with photo shop, there may be minor alterations effected when coding these screens.

* Payslip Screen
* Weekly Payroll Report Screen
* Form P30 Screen
* Form P35 Screen
* Form P60 Screen – **Is this legal?**
* Form P45 Screen

****

# Database Design

## ERD Diagram



## Database Tables (Data Types & Keys)

The Database tables are provided as a separate document titled “Payroll System Tables”

# Program Design

## Structured English/Pseudo Code

User Login

User enters username

User enters password

If username or password is invalid an error message is displayed

Only three attempts at login are allowed

Then the system will lockout

Else the system opens the main page

Add Employee

User selects add employee screen

User enters employee details

If incorrect data type is entered or data is outside the constraints an error message is displayed and correct input is requested

Else add new employee button is activated

User selects add new employee button

**Update Employee Details**

User selects update employee screen

User enters new employee details

If incorrect data type is entered or data is outside the constraints an error message is displayed and correct input is requested

Else update button is activated

User selects update button

Enter timesheet details

User selects timesheet screen

User enters timesheet details

If incorrect data type is entered or data is outside the constraints an error message is displayed and correct input is requested

Else save button is activated

User selects save button

System calculates payroll and updates database tables

**Print Payslips**

User selects payslip screen

System displays current payslip

User selects print payslips

Payslips are sent to print server

**Display Weekly Payroll Report**

User selects payroll report screen

System displays current weekly payroll report

User selects print weekly payroll report

System prints weekly payroll report

**Print Form P60**

User selects revenue report screen

User selects Form P60 tab

User selects employee name, employee number or PPS number

System displays relevant Form P60

User selects print Form P60

System prints FormP60

**Print Form P30**

User selects revenue report screen

User selects Form P30 tab

User selects P30 period (month)

System displays relevant Form P30

User selects print Form P30

System prints FormP30

**Print Form P35**

User selects revenue report screen

User selects Form P35 tab

User selects P35 period (year)

System displays relevant Form P35

User selects print Form P35

System prints FormP35

**Print Form P45**

User selects revenue report screen

User selects Form P45 tab

User selects employee name, employee number or PPS number

System displays relevant Form P45

User selects print Form P45

System prints FormP45

**User Logout**

User selects logout button

System requests logout confirmation

User selects confirm button

Logout completed

# Validation Control

Software testing should be used to examine the proper functioning of the software within the constraints of the design criteria as specified in the design documentation. Software testing should include the following,

* Expected outcome of predefined tests;
* A good test case should lead to the exposure of errors;
* To find errors during testing is not a failure but a successful test;
* These tests are independent from coding;
* The application user and programmers experience and expertise should be employed;
* Testers and coders use different tools to test the software;
* We must examine the unusual case and not only the usual case;
* Documenting the test and the test results permits its reuse and an independent confirmation of the pass/fail status of a test outcome during subsequent review.

Once the prerequisite tasks have been fully completed, software testing can begin. It starts in the small with unit testing and finishes in the large with full system testing. There may be a distinct integration level of testing. A software product should be tested to the extreme with test cases based on its internal structure and with test cases based on its external specification. The tests should provide a complete and rigorous examination, of compliance, functionality, performance, and interfacing requirements.

White box testing is also known as structural testing. It identifies test cases based on knowledge obtained from the source code, detailed design specification, and other development documents. These tests challenge the control and decisions made by the code; and the program's data structures including configuration tables. Structural testing identifies redundant code that is not executed when the program executes. Structural testing is primarily black box testing of the modules, only concerned with the inputs to the module and measuring the expected outputs.

The software requirements specification document should contain a written definition of the software functions. It is not possible to validate software without predetermined and documented user requirements.

Typical software requirements specify the following,

* System inputs;
* System outputs;
* Functions the system will perform;
* Performance requirements the system will achieve, such as data throughput, reliability, and timing;
* Interfaces,
* How will users interact with the system;
* What type of errors occur and how those errors will be handled;
* Process times;
* What type of operating environment will the software have to perform in, hardware platform or operating system;
* All ranges, limits, borders, defaults, and specified values the software must accept;

**You need to document how input will be validated and the database protected from corrupt data**

# Test Design

A complete and comprehensive set of indexed test sheets shall be generated throughout the system testing, all system test conditions, test data, expected results and actual results complete with comments shall be recorded. An example of the test control sheet to be used is included below.

## TEST CONTROL SHEET

**You need an extra column at the end for action taken, if any**

**System: Payroll System** **Prepared by: JM**

**Test Cycle: User Login** **Date:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST NO.** | **TEST CONDITION** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT/COMMENT** |
| 1 | User enters Username | User enters invalid username | Cursor moves to password textbox |  |
| 2 | User enters password | User enters valid password | Invalid login message displayed |  |
| 3 | User enters Username | User enters valid username | Cursor moves to password textbox |  |
| 4 | User enters password | User enters invalid password | Invalid login message displayed |  |
| 5 | Invalid login message displayed  **This is not a text condition** |  | Username/password textbox’s cleared, enter valid username & password message displayed |  |
| 6 | User enters username  **Are these duplications?** | User enters valid username | Curser moves to password textbox |  |
| 7 | User enters password | User enters valid password | Login button becomes active |  |
| 8 | User selects Login button |  | Main page opens |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |

# You need a lot more test cases than this