B

**A2 Computing Project**

**Building Job Income and Outgoings Tracker**

****

**Ted Eriksson**

Contents

[Analysis 5](#_Toc289934437)

[Background 5](#_Toc289934438)

[Current system 5](#_Toc289934439)

[Identification of User 5](#_Toc289934440)

[Problem definition 5](#_Toc289934441)

[Impact od Problems 6](#_Toc289934442)

[Feasibility analysis 6](#_Toc289934443)

[Fact finding 7](#_Toc289934444)

[Requirements analysis 8](#_Toc289934445)

[Requirements specification 9](#_Toc289934446)

[DFDs 10](#_Toc289934447)

[Level 0 10](#_Toc289934448)

[Level 1 11](#_Toc289934449)

[ERDs 12](#_Toc289934450)

[Research of alternative solutions 12](#_Toc289934451)

[Realistic appraisal of feasibility 12](#_Toc289934452)

[Justification of chosen solution 12](#_Toc289934453)

[Agreed system objectives and scope 12](#_Toc289934454)

[Design 13](#_Toc289934455)

[Outline system design 13](#_Toc289934456)

[System flow chart 14](#_Toc289934457)

[User interface design 15](#_Toc289934458)

[UI Overview 15](#_Toc289934459)

[Login and Daily Form 16](#_Toc289934460)

[New Job 17](#_Toc289934461)

[Results 18](#_Toc289934462)

[Edit Details 19](#_Toc289934463)

[Hardware specification 20](#_Toc289934464)

[Program structure 20](#_Toc289934465)

[Hierarchy charts 20](#_Toc289934466)

[Structure charts 20](#_Toc289934467)

[Design data dictionary 21](#_Toc289934468)

[Object diagrams and class definitions 23](#_Toc289934469)

[Data structures 23](#_Toc289934470)

[File organisation 23](#_Toc289934471)

[Entity-Relationship diagram 23](#_Toc289934472)

[Normalised database tables 23](#_Toc289934473)

[Algorithms 23](#_Toc289934474)

[Queries 23](#_Toc289934475)

[Detailed design of printed output 23](#_Toc289934476)

[Preliminary test plan 23](#_Toc289934477)

[Detailed test data 23](#_Toc289934478)

[Implementation 23](#_Toc289934479)

[Coding 23](#_Toc289934480)

[Testing 24](#_Toc289934481)

[Test strategy 24](#_Toc289934482)

[Test plan 24](#_Toc289934483)

[Test evaluation 24](#_Toc289934484)

[Test evidence/results 24](#_Toc289934485)

[Number Field Validation 24](#_Toc289934486)

[Combo Box Fix 25](#_Toc289934487)

[Implementation 26](#_Toc289934488)

[Maintenance 26](#_Toc289934489)

[User Guide 26](#_Toc289934490)

[Appraisal 26](#_Toc289934491)

[Appendices 26](#_Toc289934492)

[Interview transcripts 26](#_Toc289934493)

[Questionnaires 26](#_Toc289934494)

[IT.1 - COMPANY Background Questionaire 26](#_Toc289934495)

[Original system documents 28](#_Toc289934496)

[OSD.1 28](#_Toc289934497)

[Source code 29](#_Toc289934498)

[Test data 29](#_Toc289934499)

[Acknowledgements 29](#_Toc289934500)

# Analysis

## Background

The company was founded on January 1st 1995, and later taken into a limited company in March 2006. The company currently has two full time employees and one to fifteen subcontractors at a time. The annual turnover for the company is approximately £250,000. It is based near Hailsham, East Sussex. The business provides general building work from a kitchen to oak-framed barns.

## Current system

At present Eriksson Building Services ltd use spreadsheets to work out the pay for each worker and the income and outgoings for the business. See OSD.1 in the appendices.

Eriksson Building Services ltd employs up to 15 people at any one time, at the end of the day each workers’ hours and the task they were doing must be logged and stored in a database. Also any material bought for the jobs must be logged as well as where it was bought from so as to work out the running costs as well as the quarterly tax returns for the business.

|  |  |
| --- | --- |
| Pros | Cons |
| The user knows how to use this system  Spreadsheets are easily printable | They take a long time to fill out  Information has to be inputted several times in order for everything to correspond |

## Identification of User

My User is the owner of Eriksson Building Services ltd. He has limited skills on a computer, but is capable of using programs with easy to follow graphical user interfaces. He has no understanding of Microsoft Access or any other similar program at all.

## Problem definition

Currently the company uses spreadsheets to input this data; this has to be done manually and requires some training in order for them to use it correctly. If information is inputted incorrectly then financial errors could be made causing a customer to lose their faith in the company therefore damaging the company’s reputation. If the user does have training, there is still a high chance for human error, so each record is checked for mistakes. This is very time consuming and is a problem especially when the company needs to keep to tight deadlines.

### Impact od Problems

What’s the problem with the current system?

* Using spreadsheets
* They have to be completed manually
* They require training to be of any use
* Only specific people with the training can get information to and from them
* Information could be inputted incorrectly
* Company records will be incorrect
* Financial errors made
* Customer loses faith in company
* Company reputation is tarnished

So what if the user has the training?

* It still takes a long amount of time to input data into spreadsheets
* User has to spend time checking records for errors
* This time could be better spent doing other work for the customer
* Deadline missed to inadequate system
* Customer dissatisfied
* Company reputation is tarnished

## Feasibility analysis

The problem is better solved by a computer because:

* if the system was paper based data could easily be lost
* writing data down would be a very long laborious task
* hard to backup the data

A new system would be feasible because:

* currently the repetitive nature of putting data into the spreadsheets takes a larger amount of time than it would with a more streamlined system

This problem is one that is appropriate to be solved by a computer-based solution, because it relies on being able to fetch from a database of hours, costs and names frequently and rapidly. If this was done manually the user would have to search through all their files manually, this would be very time consuming and also, will use considerable storage space (i.e. filing cabinets).

Furthermore, backing up hand written data would be an extremely tedious task and would result in loss of working hours for the user. If there was a fire all data would be lost, where as if the system was computer based, a copy of the database could be made in seconds and kept on a flash drive elsewhere. A bespoke programmed solution would be significantly more beneficial than the existing spreadsheet system because time is an important factor for the user, as time spent inputting data into a computer means less time working and therefore a loss in earnings. While spreadsheets are good in some respects such as being easy to back-up, in others they are not, such as: They take a long time to fill out and can be hard for a new user to become competent.

The Problem with using spreadsheets compared with a database is the fact that a spreadsheet is only in two-dimensions, meaning you can only compare one thing against another at one time (e.g. comparing hours and time). A database however can have filters placed upon it meaning that you can compare hours by time WHEN Total Pay is > £10. This gives the user much more flexibility and can be very useful if they want to find specific data quickly.

## Fact finding

I compiled a questionnaire to find out more about the company background (see IT.1 in the appendices)

I also had an informal and formal meeting to discuss the current system and problem definition.

## Requirements analysis

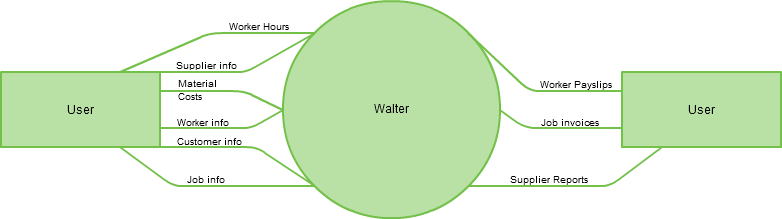
## Requirements specification

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Reference Key: | |  | | F | | Functional | | S | Security | | | Rel | | Reliability | | P | | Performance | | R | | Reporting | | Pr | | Processing | | |  |  | | --- | --- | | Priority Key: |  | | M | Must | | W | Would Like | | C | Could | | S | Should | | |  |  | | --- | --- | | Source Key: |  | | FML2 | Formal Meeting 2 | | FML3 | Formal Meeting 3 | |

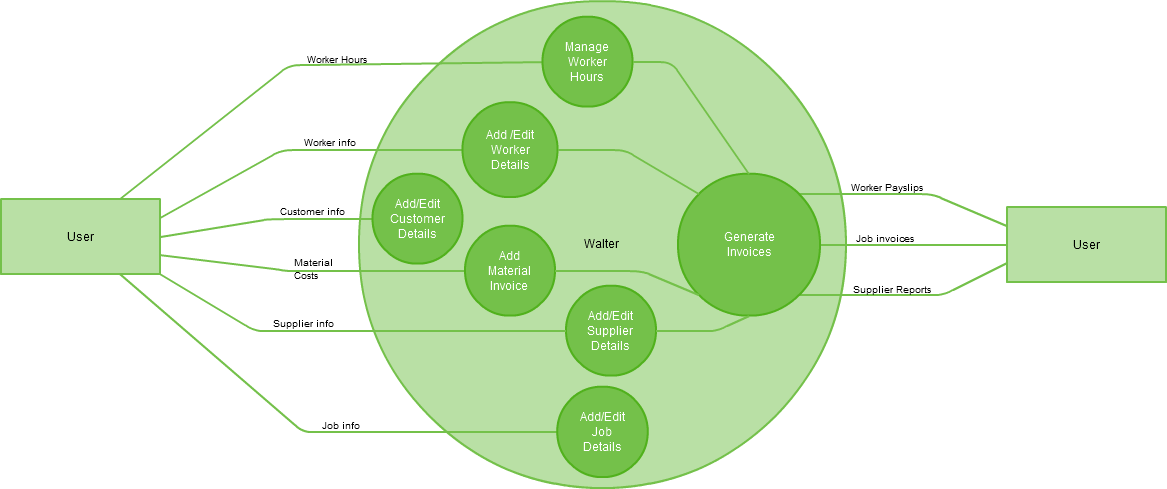
|  |  |  |  |
| --- | --- | --- | --- |
| Reference | Requirement | Source | Priority |
| F1 | The system shall be able to use a drop down list for workers within the daily data form | FML2 | M |
| F2 | The system shall be able to input the amount of hours worked by a worker | FML2 | M |
| F3 | The system shall be able to input the wage of each worker | FML2 | M |
| F4 | The system shall be able to remember the details (name, address) of each worker | FML2 | M |
| F5 | The system shall be able to input what the worker was doing on a specific day | FML2 | M |
| F6 | The system shall be able to input the cost of materials bought for a job | FML2 | M |
| F7 | The system shall be able to store the details of a Customer such as Customer name and address | FML2 | M |
| F8 | The system shall be able to store the details of a Job such as Job name and address | FML2 | M |
| F9 | The system shall be able to store the details of a Supplier such as Supplier name and address | FML2 | M |
| R1 | The system shall be able to easily locate all past data inputted into the system | FML2 | M |
| R2 | The system shall be able to view previous data in an easy to read format | FML2 | M |
| R3 | The system shall be able to view previous data between two dates | FML2 | M |
| R4 | The system shall be able to view Job, Worker and Supplier data separately | FML2 | M |
| Pr1 | The system shall be able to to export data to a spreadsheet program (.xls) | FML2 | W |
| F10 | The system shall be able to log material costs separately from workers | FML3 | M |
| R5 | The system shall be able to produce a report containing information about the progress of a job | FML3 | M |
| R6 | The system shall be able to produce an invoice for a customer | FML3 | M |
| P1 | The system shall be able to load in 5 seconds of icon click | FML3 | C |
| S1 | The system shall be Password Protected to stop intruders | FML2 | M |
| Rel1 | The system shall have the correct validation as to stop invalid data being inputted into the database | FML3 | M |
| R6 | The system shall be able to show the total profit of each job | FML3 | S |

## DFDs

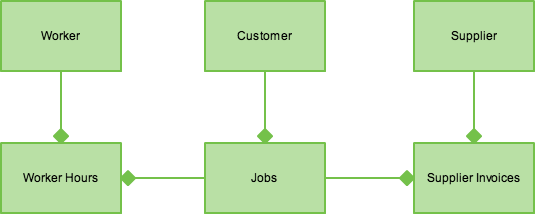
### Level 0



### Level 1



## ERDs



## Research of alternative solutions

Spreadsheet Software: This is sometimes a very good solution but spreadsheets can be confusing to new users and difficult and time consuming to use.

A paper based solution: This would be very impractical as it would require constant maintenance as well as taking up a massive amount of room as every item would have to be stored. Also it would be incredibly difficult to back up.

Business Management Software: These are powerful programs, but while they have a lot of functionality, they are expensive. Also in a bid to make them seem easier to use they do a lot of work behind the scenes, which can make them difficult for some users to trust.

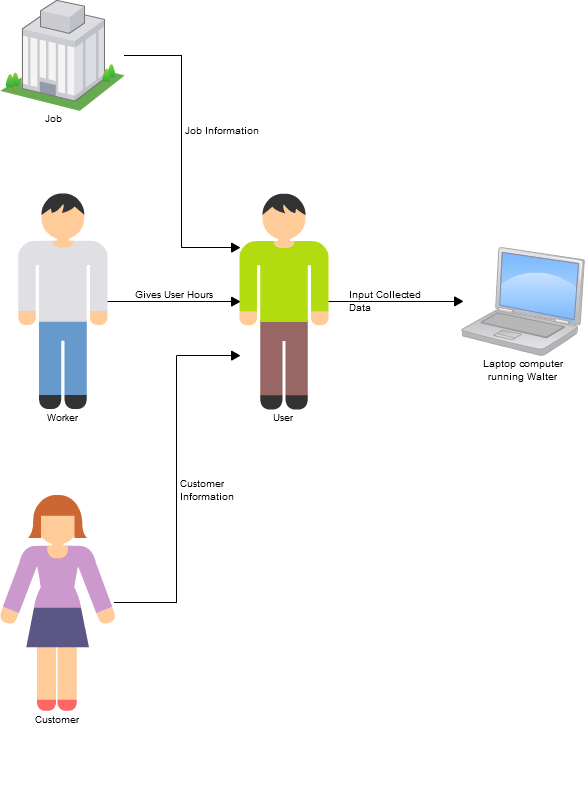
## Realistic appraisal of feasibility

## Justification of chosen solution

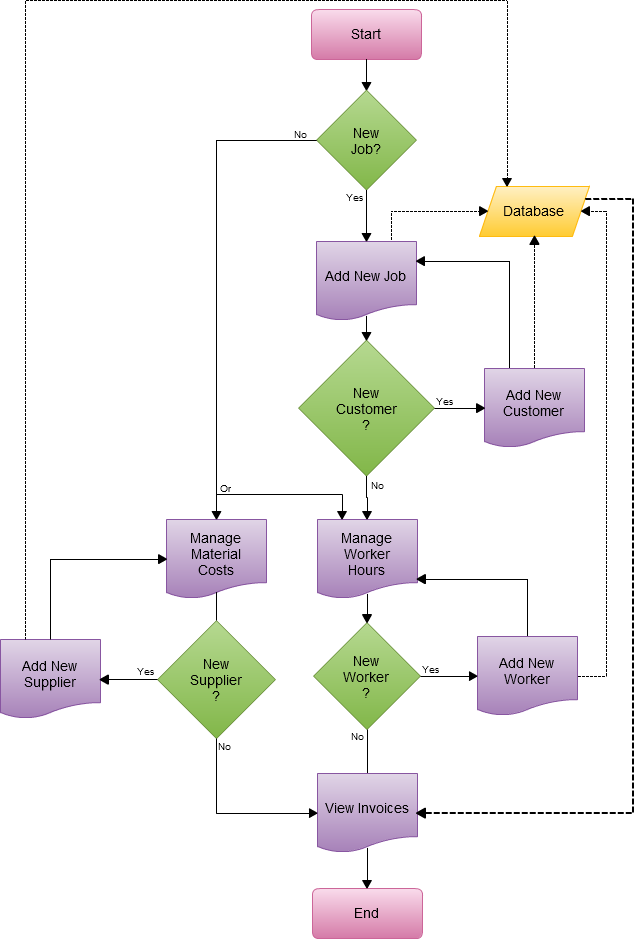
## Agreed system objectives and scope

# Design

## Outline system design

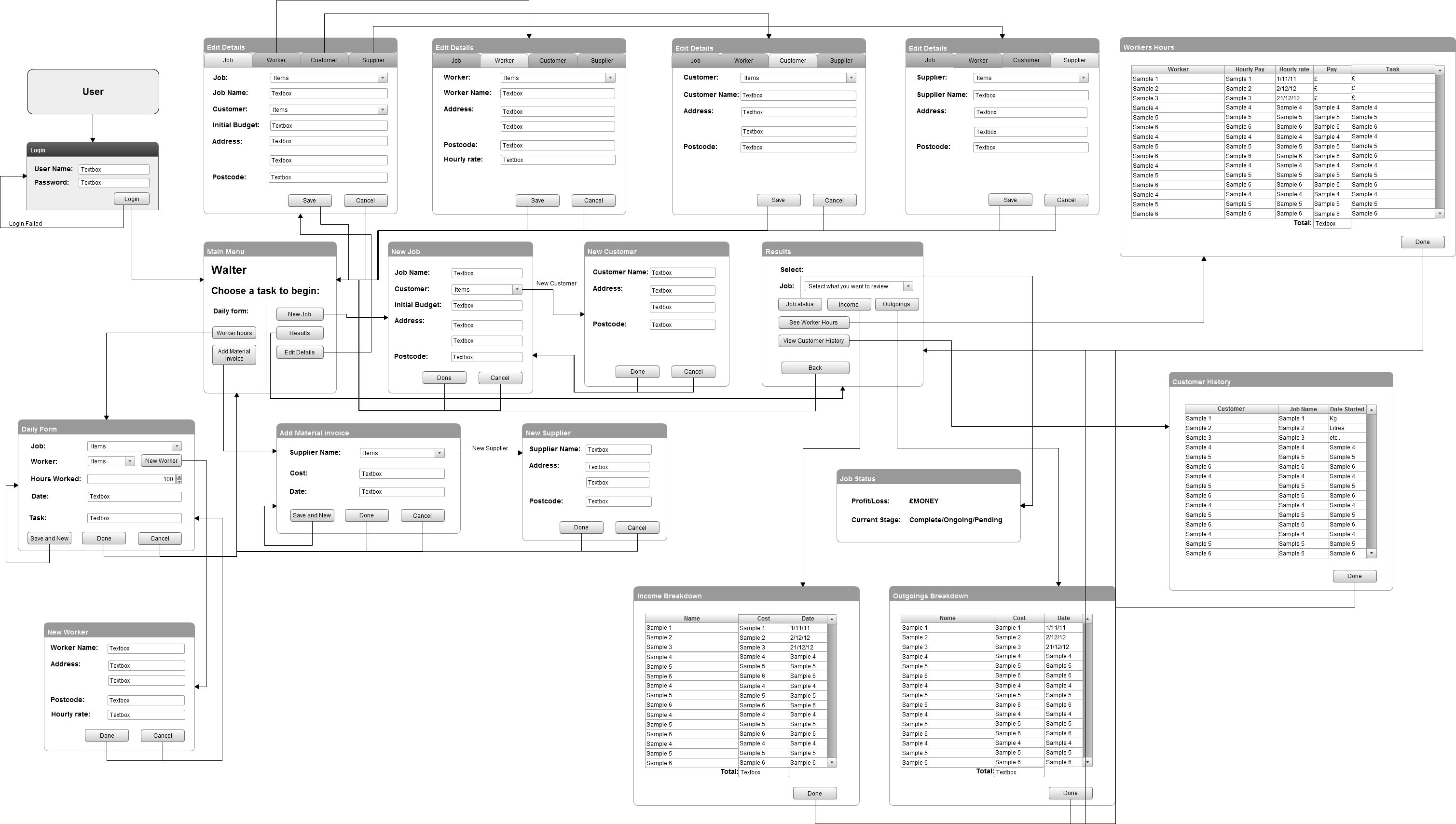


## System flow chart



## User interface design

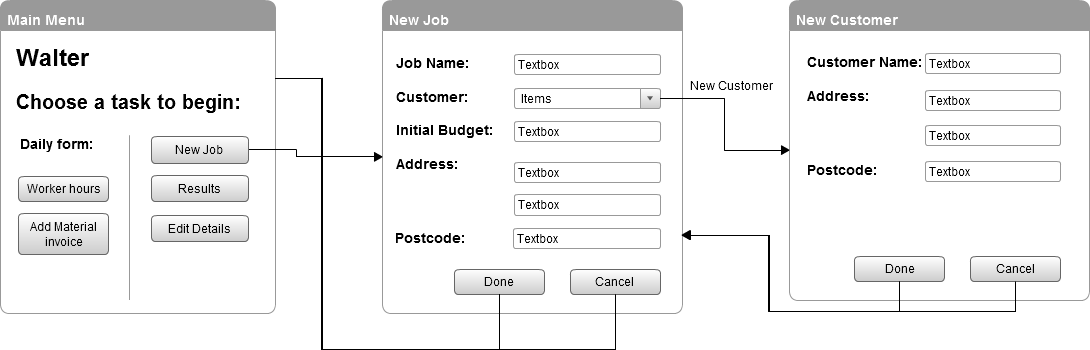
### UI Overview



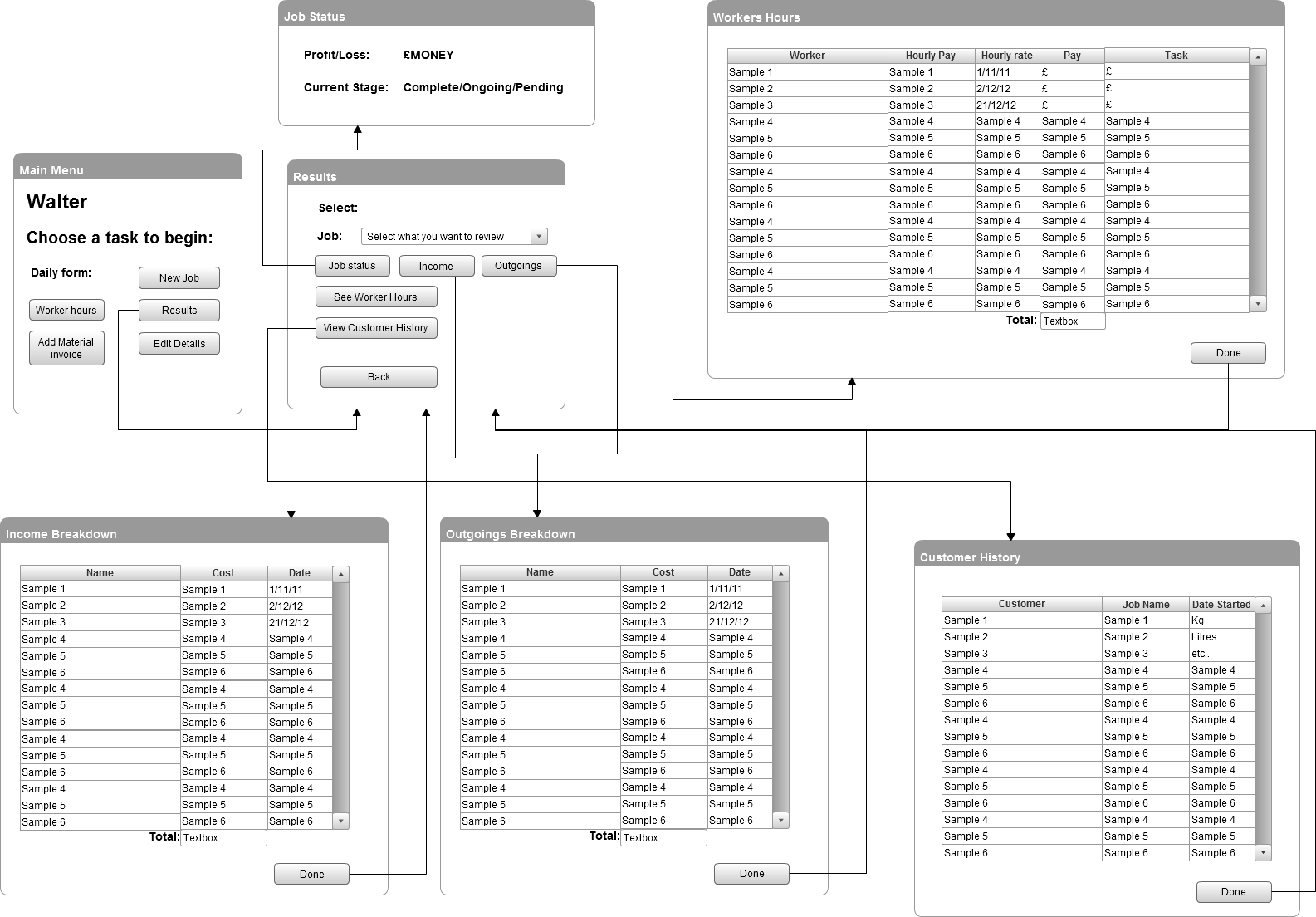
### Login and Daily Form



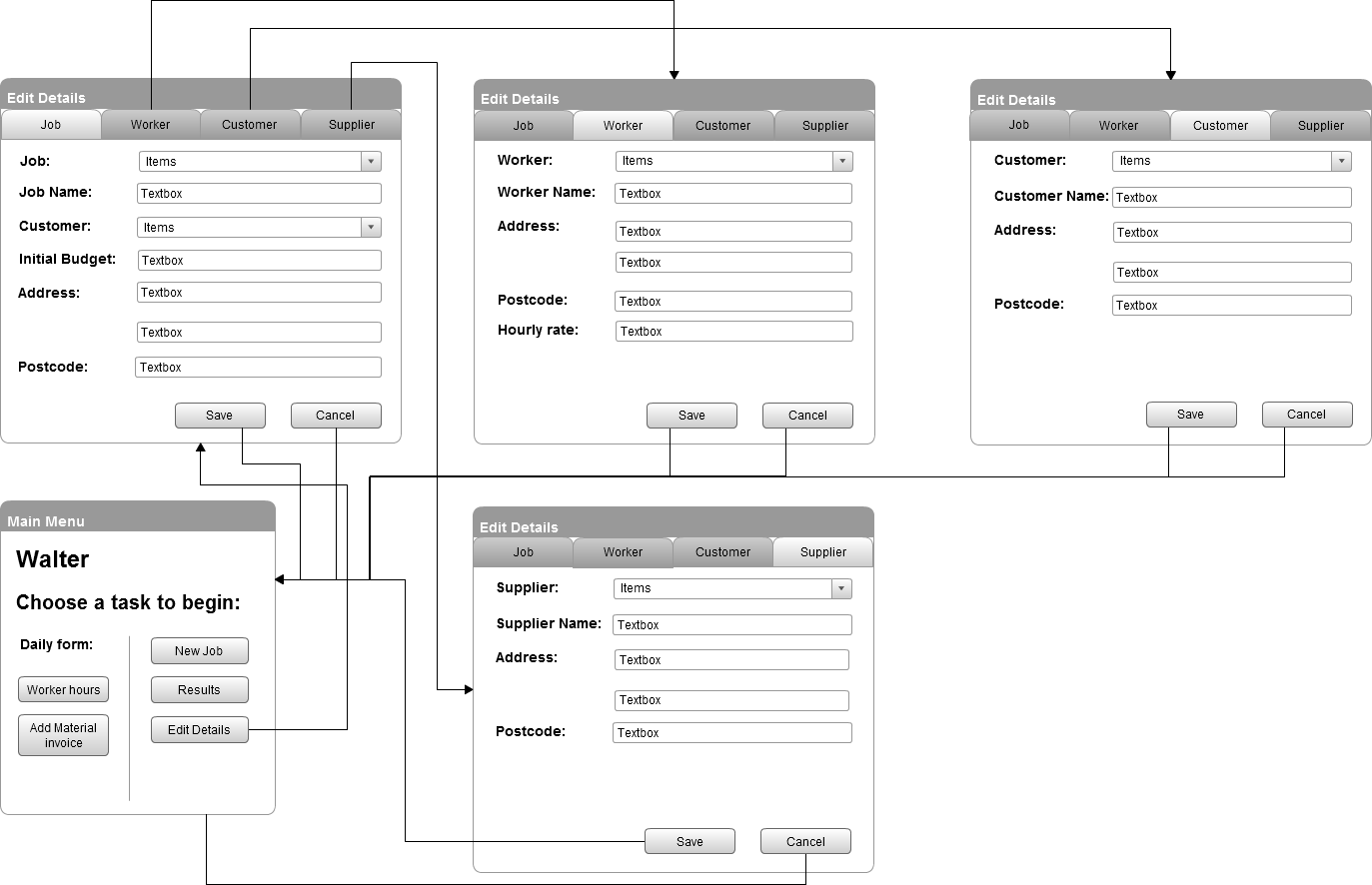
### New Job



### Results



### Edit Details



## Hardware specification

|  |
| --- |
| System Requirements |
| O/S: Windows XP or Above  1GHz Processor  1 GB RAM  30 MB + Hard drive space |

## Program structure

## Hierarchy charts

## Structure charts

## Design data dictionary

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table | Fieldname | Contents | Type | Format | Range | Required | PK/FK | Reference |
| Jobs | Job\_ID  Job\_Name  CustomerID  Initial\_Budget  Address1  Address2  Postcode | ID for a job  Name of the job  ID for a Customer  Initial budget for a job  Address line 1 for a job  Address line 2 for a job  Postcode of a jobs address | Number  Text  Number  Currency  Text  Text  Text |  |  |  | PK |  |
| Customer | CustomerID  Customer\_Name  Address1  Address2  Postcode | ID for a customer  Name of a Customer  Address line 1 for a customer  Address line 2 for a customer  Postcode for a customer | Number  Text  Text  Text  Text |  |  |  | PK |  |
| Worker | WorkerID  Worker\_Name  Address1  Address2  Postcode  Hourly\_Rate | ID for Worker  Name of Worker  Address of Worker  Address of Worker  Postcode of Worker  Amount a Worker earns per hour | Number  Text  Text  Text  Text  Text |  |  |  | PK |  |
| Worker\_hours | ID  WorkerID  JobID  Date\_worked  Hours\_worked | ID for Record  ID for Worker  ID for a job  Date a Worker worked  Amount of hours worked | Number  Number  Number  Date/Time  Number |  |  |  | PK  FK  FK |  |

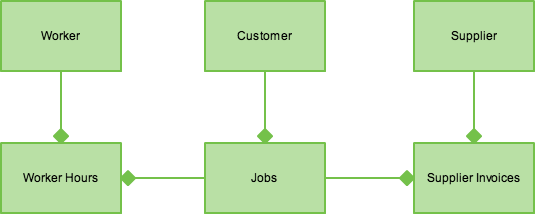
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table | Fieldname | Contents | Type | Format | Range | Required | PK/FK | Reference |
| Supplier | SupplierID  Supplier\_Name  Address1  Address2  Postcode | ID for Supplier  Name of a Supplier  Address of Supplier  Address of Supplier  Postcode of Supplier | Number  Text  Text  Text  Text |  |  |  | PK |  |
| Supplier\_outgoings | ID  SupplierID  JobID  Cost  Date\_of\_invoice | ID of Record  ID for Supplier  ID for a job  Cost of Materials  Date of Suppliers invoice | Number  Number  Number  Currency  Date/Time |  |  |  | PK  FK  FK |  |
| Options | User  Pass | ID for the user of the database  Password for the users account | Text  Text |  |  |  | PK |  |

## Object diagrams and class definitions

## Data structures

## File organisation

## Entity-Relationship diagram



## Normalised database tables

## Algorithms

## Queries

## Detailed design of printed output

## Preliminary test plan

## Detailed test data

# Implementation

## Coding

# Testing

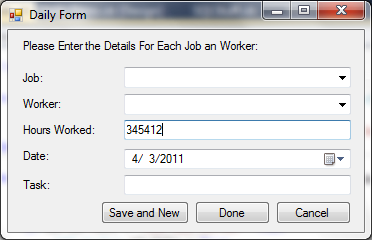
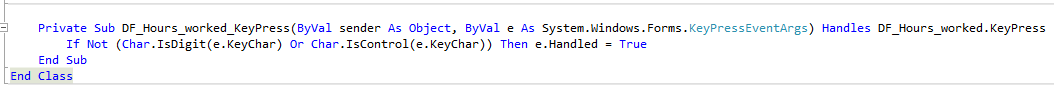
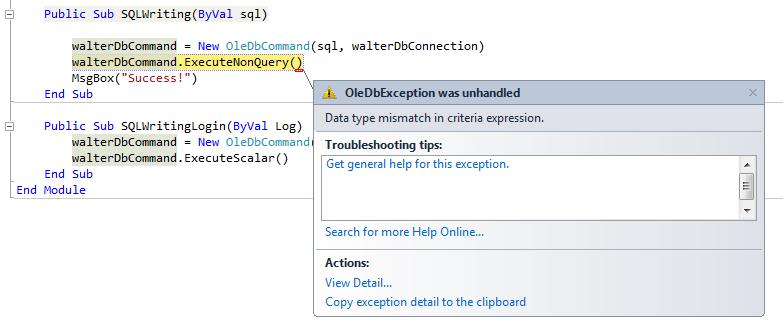
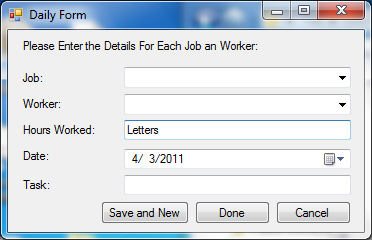
## Test strategy

## Test plan

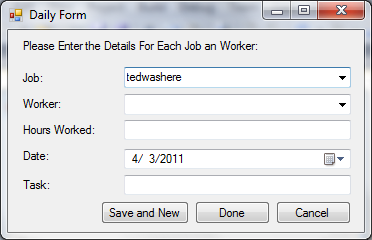
## Test evaluation

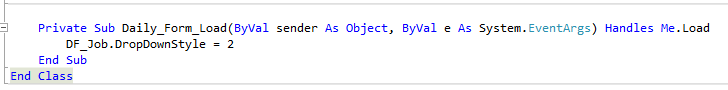
## Test evidence/results

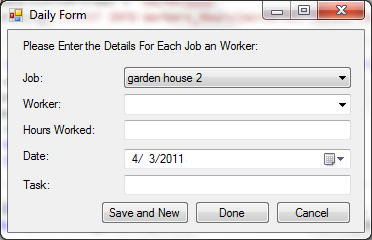
### Number Field Validation



### Combo Box Fix







# Implementation

# Maintenance

# User Guide

# Appraisal

# Appendices

## Interview transcripts

## Questionnaires

## IT.1 - COMPANY Background Questionaire

#### Q: When and how was the company established?

A: It was a partnership that started January 1st 1995, and taken into a limited company in March 2006.

#### Q: How many employees are currently working for Eriksson Building Services LTD?

A: Only two employees, the rest (1-15 at a time) are sub contractors.

#### Q: What is the annual Turnover for the company?

A: Approximately £250,000

#### Q: Where is the company based?

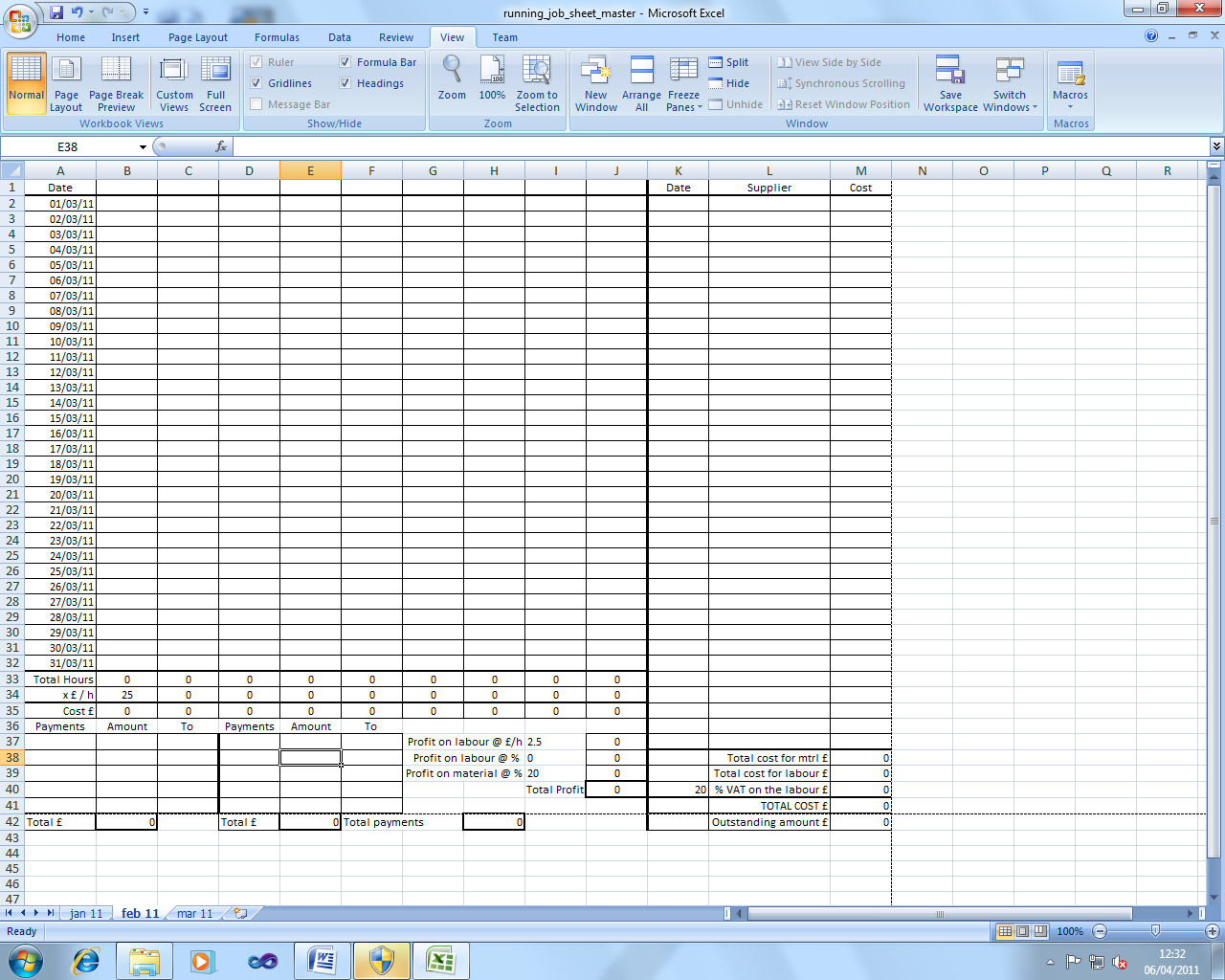
A: the company is based at apple barn, old road, magham down, bn27 1pr

#### Q: In a nutshell, what service does the company provide?

A: General building services, anything from a kitchen to an oak-framed barn.

## Original system documents

### OSD.1

This is the main Spreadsheet for working out the cost of a job.

## Source code

## Test data

## Acknowledgements