

AY Patel 27342514

RB De klerk 25007335

Nj roberts 25994689

fj verster 25021516

A PRETORIUS 26056569

N PRETORIUS 27532704

Analysis paralysis

ITRW 225

Table of Contents

[Project Plan: 1](#_Toc491872466)

[Feasibility Matrix & Cost Benefit Analysis 1](#_Toc491872467)

[Physical Data Model 5](#_Toc491872468)

[Physical Process Model 6](#_Toc491872469)

[File/Server Architecture DFD 7](#_Toc491872470)

[Prototypes of Input, Output & User Interface 7](#_Toc491872471)

[Database Schema 11](#_Toc491872472)

[Example Code 12](#_Toc491872473)

[Example Reports 13](#_Toc491872474)

[Figure 1 - Login 7](#_Toc491877498)

[Figure 2 - Main Form 8](#_Toc491877499)

[Figure 3 - New Booking Form 8](#_Toc491877500)

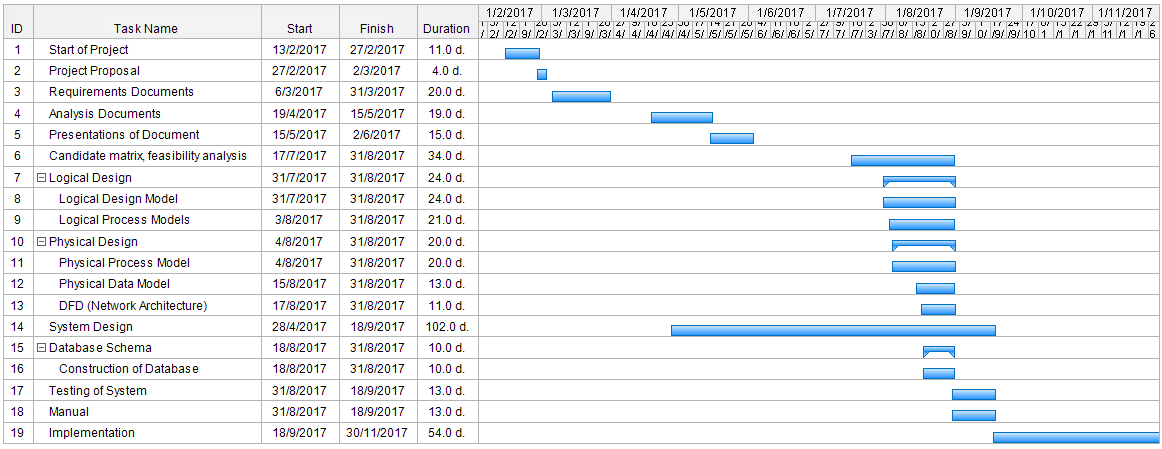
[Figure 4 - Review Scheduled Booking 9](#_Toc491877501)

[Figure 5 - Finalize Booking (Payment) 9](#_Toc491877502)

[Figure 6 -Adding of new Expenses 10](#_Toc491877503)

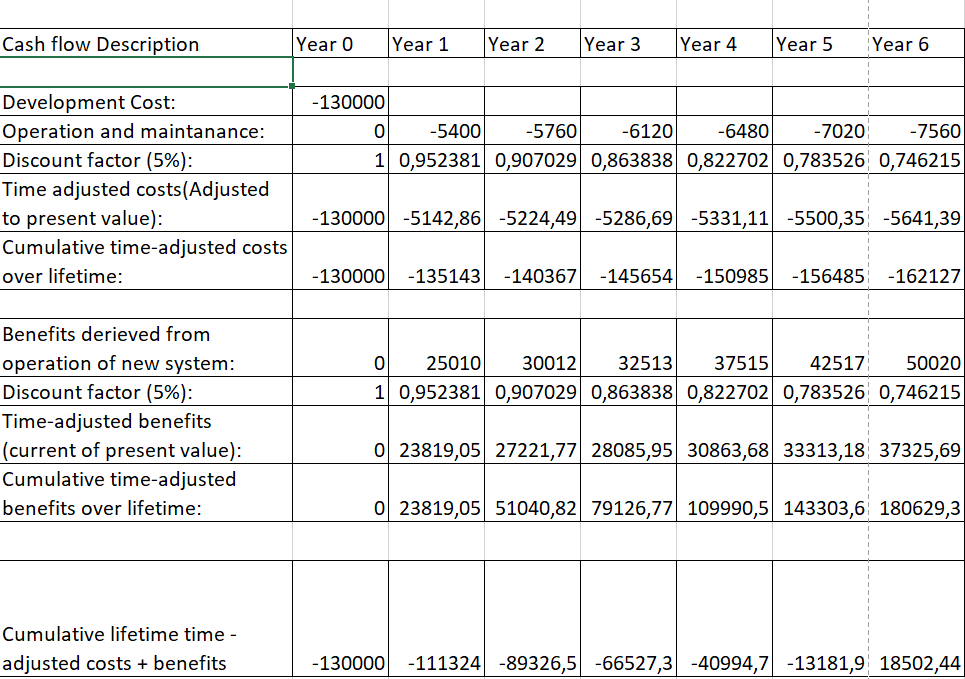
[Figure 7 - Paid Expenses 10](#_Toc491877504)

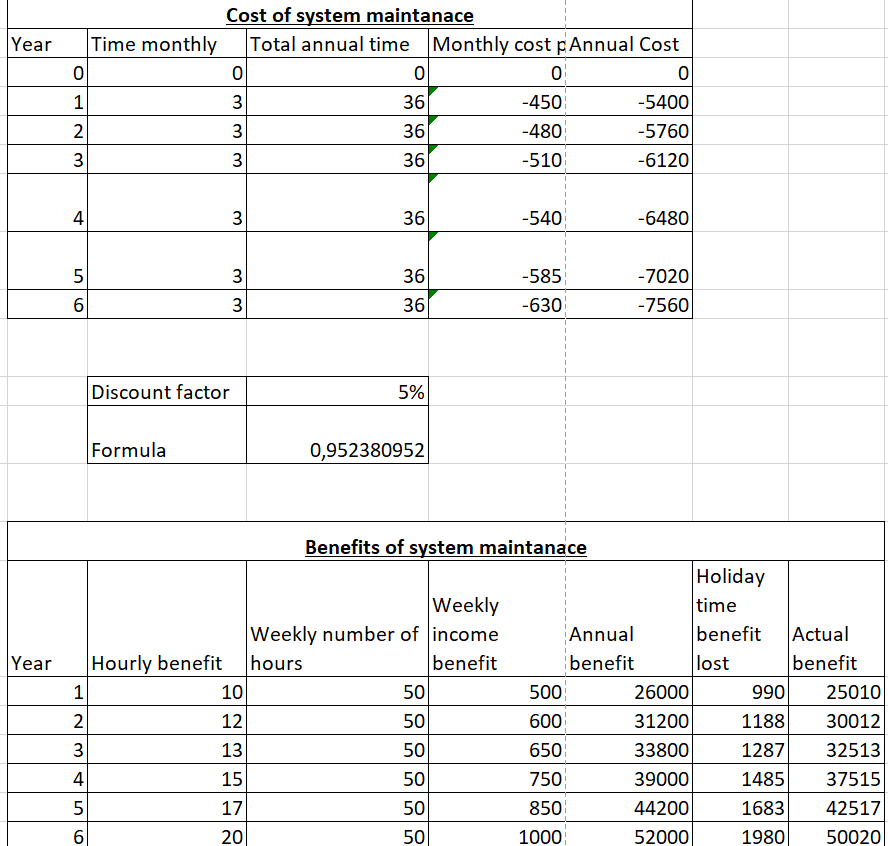
# Project Plan:



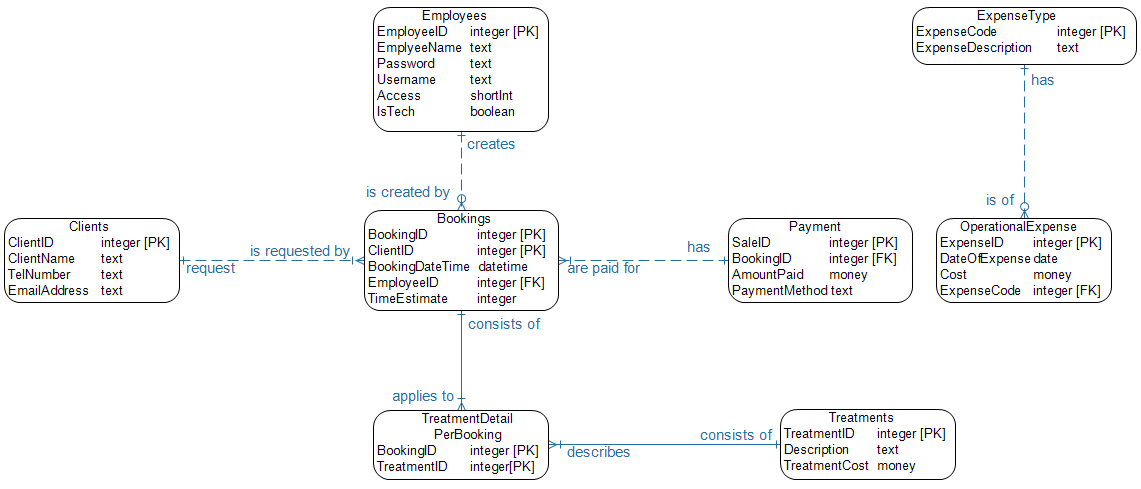
# Feasibility Matrix & Cost Benefit Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Weighting** | **Candidate 1** | **Candidate 2** | **Candidate 3** |
| **Description** |  | New in-house development  using C# and MS  Access | New in-house development  using C++ and  SQL | Same as candidate 2 |
| **Operational feasibility** | 15% | Client requirements and functionality are  fully supported.  Score: 100 | Support for most of client  requirements and some functionality.  Concerns about  ease of use. Score: 60 | Supports minimal  requirements. Unsuited for functionality required. Score: 30 |
| **Cultural Feasibility** | 15% | No foreseeable  problems. Score : 100 | Alternate interface may not  suit user. Score  70 | No foreseeable  problems. Score : 100 |
| **Technical feasibility** | 20% | Works well, exceptions include minor bugs in  input, unexpected crashes, long startup time.  Score: 80 | Works flawlessly, no bugs but may crash under strain  Score: 90 | Works perfectly. Score 100 |
| **Schedule feasibility** | 10% | Less than year. Score: 80 | 18 - 30 months. Score: 50 | 9 months. Score: 100 |
| **Economic feasibility** | 30% |  |  |  |
| **Cost to develop:** |  | R216 000 | R380 000 | R120 000 |
| **Payback (discounted)** |  | 48 months | 3 years | Within 1 year |
| **Net present value: (After 6 years)** |  | R50 020 | Approx: R30 000 | Approx: R10 000 |
| **Detailed calculation** |  | See cost benefit analysis | - | - |
| **Legal feasibility** | 10% | No foreseeable problems  Score: 100 | No foreseeable problems  Score: 100 | No foreseeable problems  Score: 100 |
| **Weighted score** | 100% | 91% | 65% | 87% |

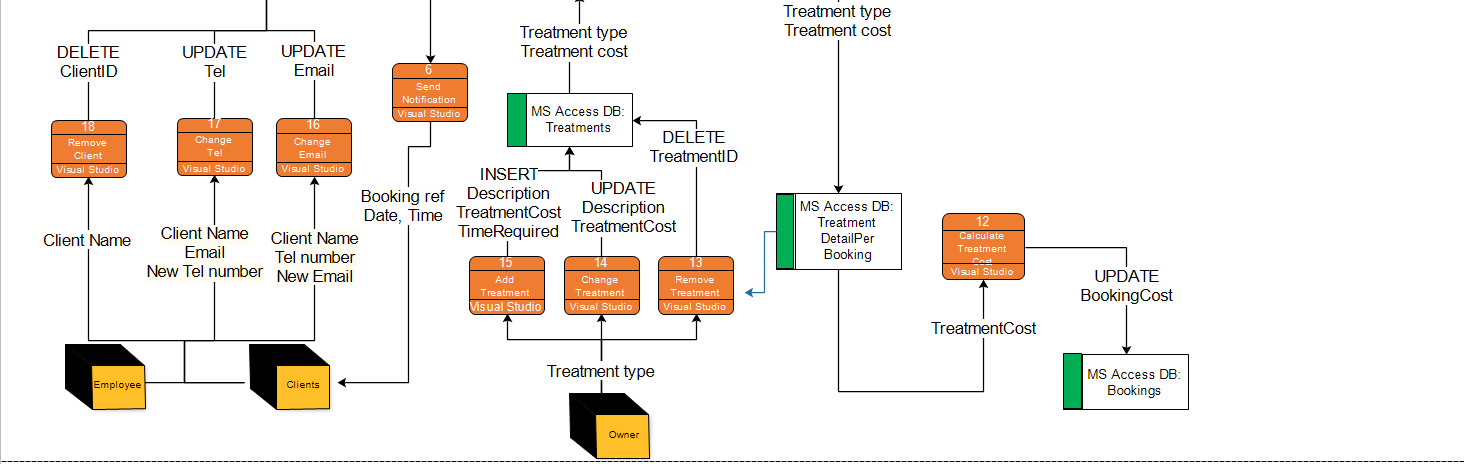
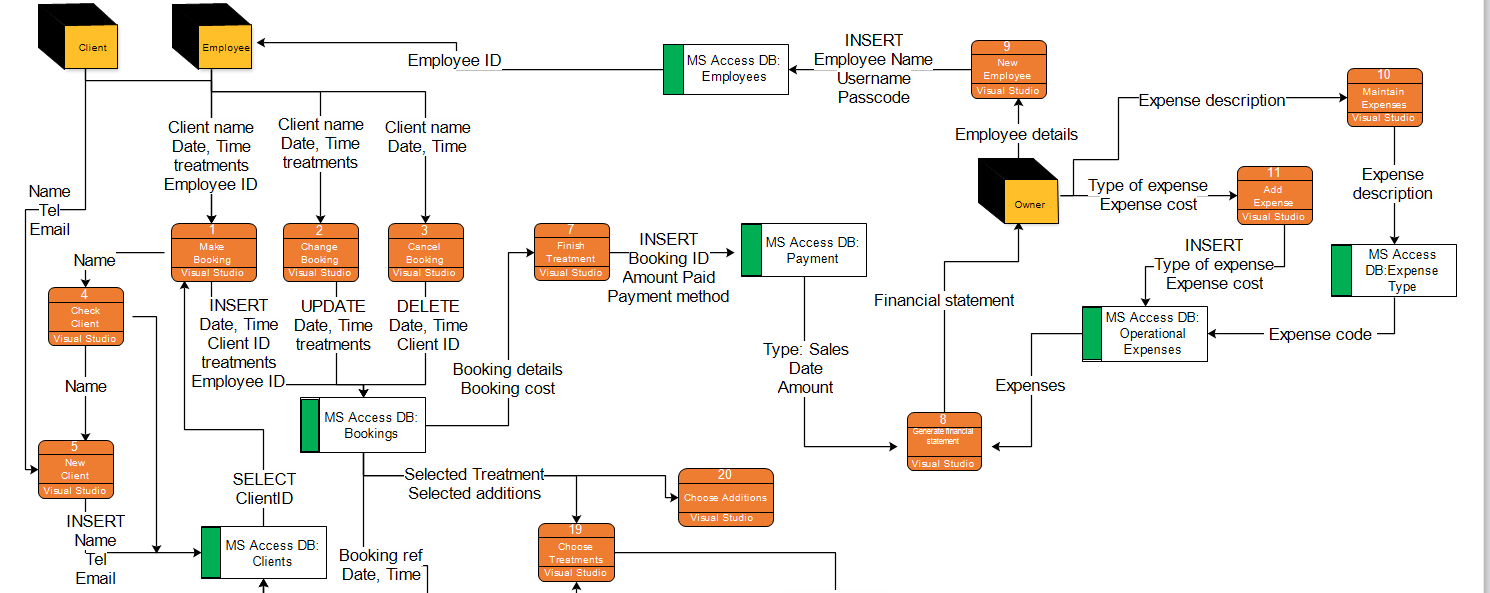


z

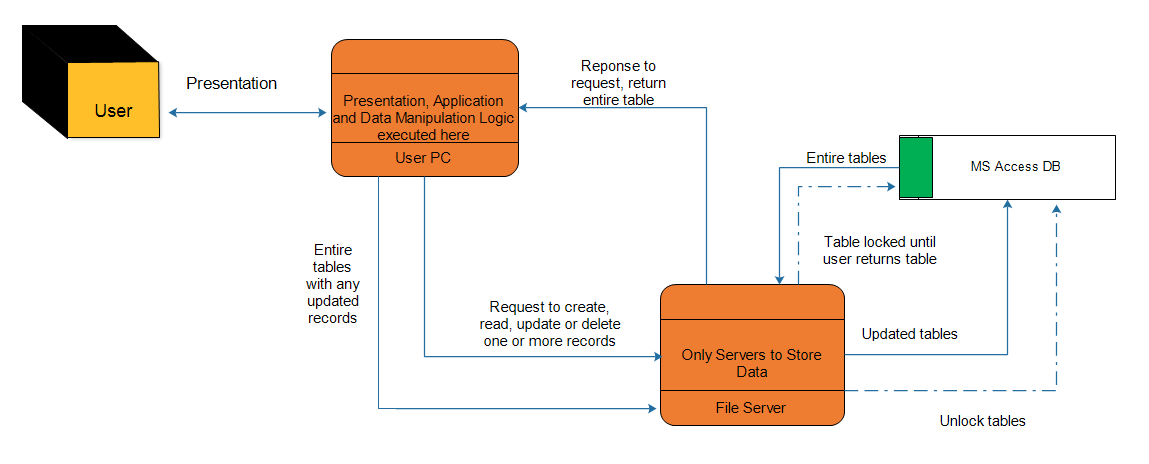
# Physical Data Model



# Physical Process Model



# File/Server Architecture DFD



# Prototypes of Input, Output & User Interface

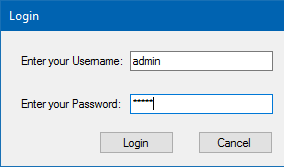


Figure - Login

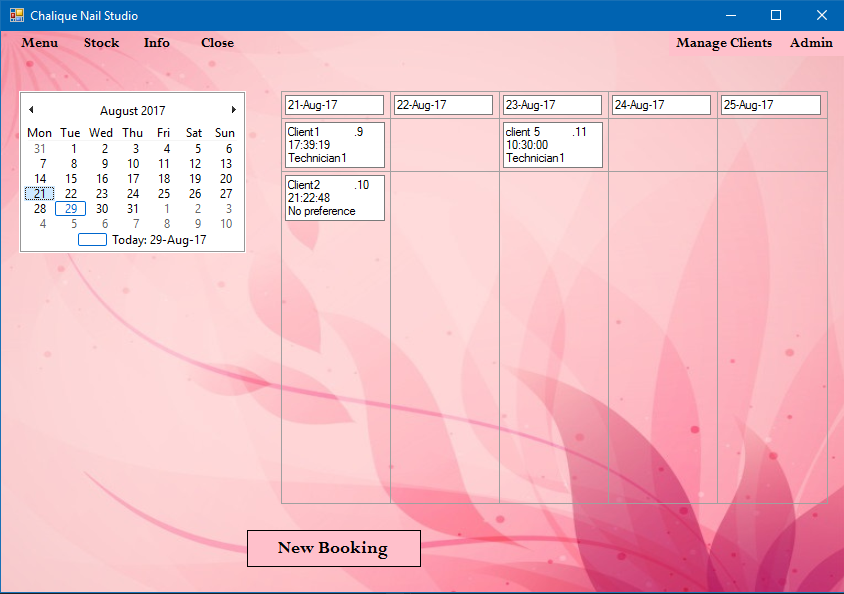


Figure - Main Form

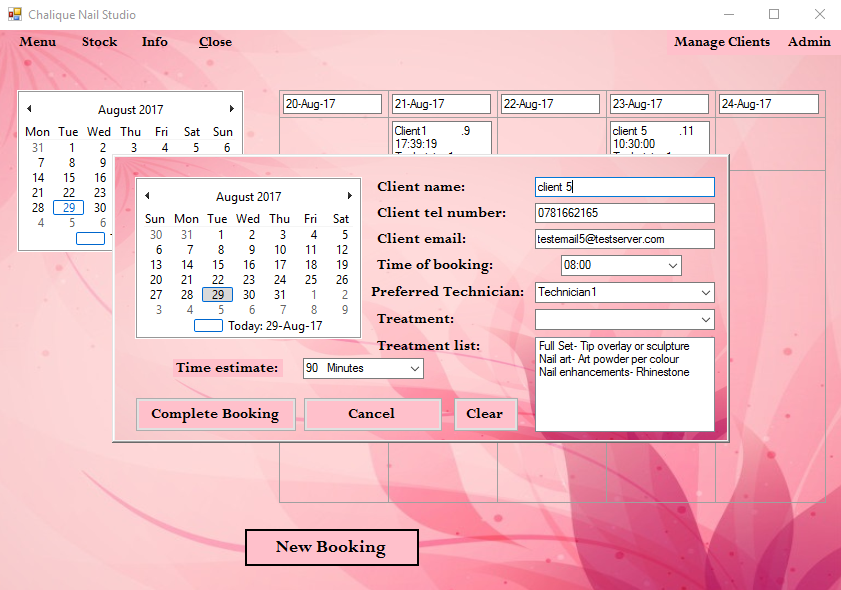


Figure - New Booking Form

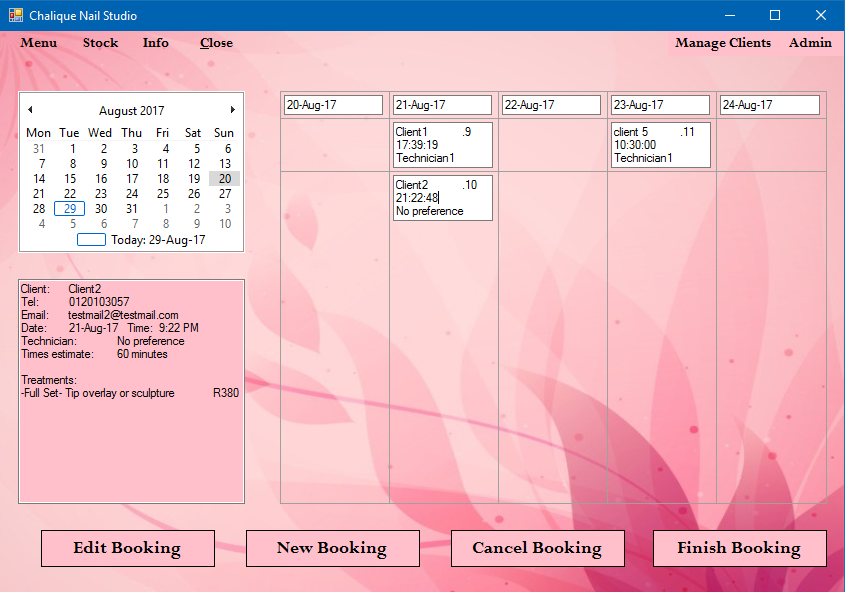


Figure - Review Scheduled Booking



Figure - Finalize Booking (Payment)

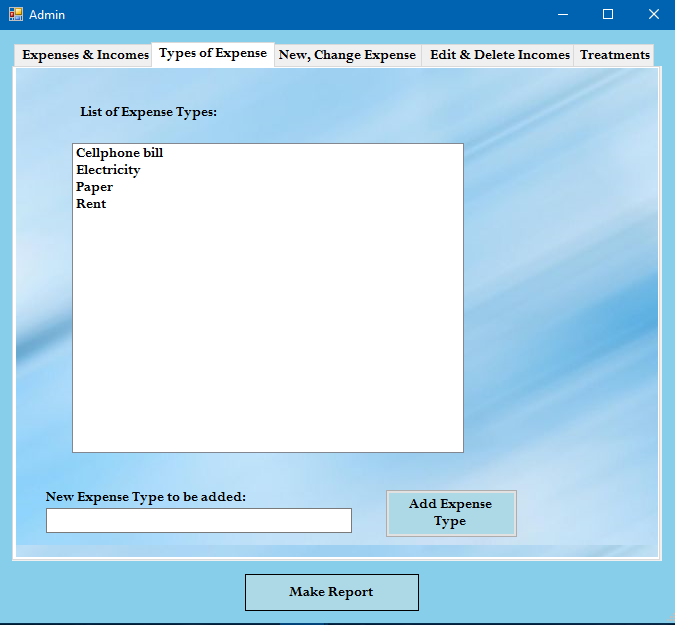


Figure -Adding of new Expenses

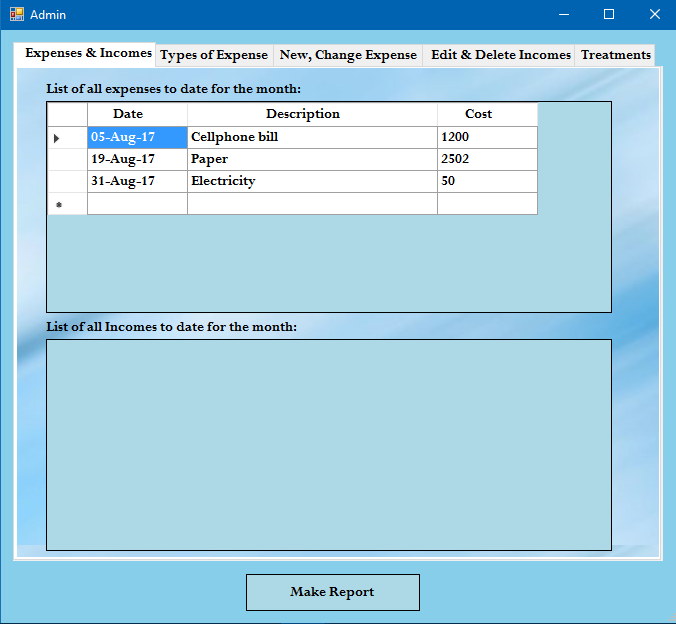
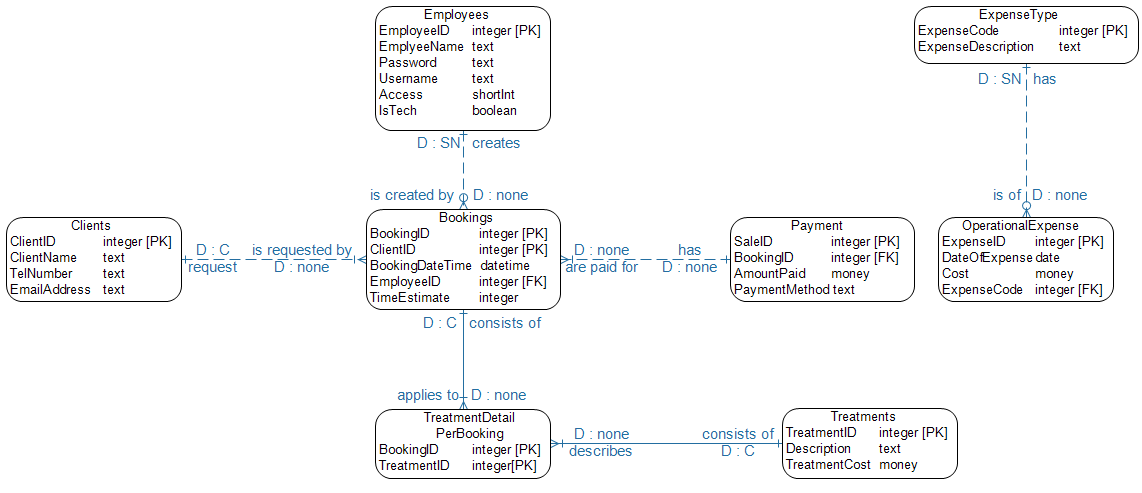


Figure - Paid Expenses

# Database Schema



# Example Code

//Button click adds an expense into the database, Calls method that displays the expense in the listbox on tab page 2 as well

private void btnAddExpenseType\_Click(object sender, EventArgs e)

{

string expType = txtExpenseType.Text;

//check if the textBox is empty or not before adding information in the database

if(string.IsNullOrEmpty(expType))

{

MessageBox.Show("Please enter an expense type!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

if(MessageBox.Show("Are you sure you would like to add " + expType + " as an expense?", "Request", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

try

{

//Opens the DB connection

conn.Open();

//Command to insert expense type into Database

OleDbCommand cmd = new OleDbCommand($@"INSERT INTO ExpenseType(ExpenseDescript) VALUES ('{expType}')", conn);

cmd.ExecuteNonQuery();

//Expense type that is added is shown in messageBox

MessageBox.Show("You have successfully added " + expType + " as an expense", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);

//Close DB connection

conn.Close();

//Calls method that reads the database and displays the expense type in the listBox

readExpenseTypeToListbox();

//Clears the contents of the textBox

txtExpenseType.Clear();

}

catch(Exception ex)

{

MessageBox.Show("There was an error\n" + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show(expType + " was not added", "Declined", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

txtExpenseType.Focus();

}

# Example Report

