Institiúid Teicneolaíochta Cheatharlach

INSTITUTE *of* TECHNOLOGY CARLOW

At the Heart of South Leinster

Design Manual

For

iPhone Application

Submission Date: -

15th of January 2010

Prepared by: -

Tuna Erdurmaz (C00115609)

Supervisor: -

Paul Barry

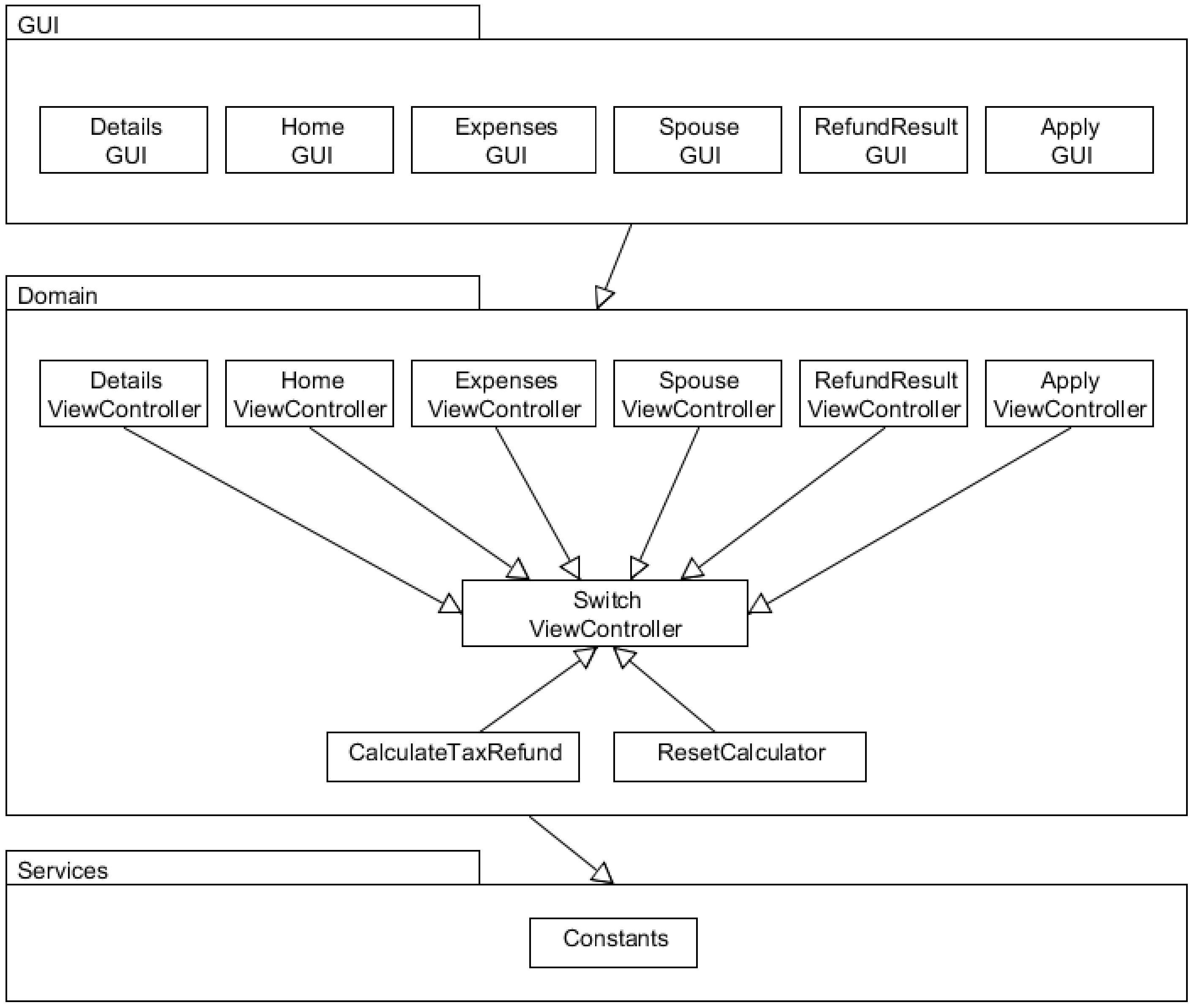
Table of Contents

1. Introduction

The purpose of this document is to make an iPhone app project easier to code, test, debug and maintain. This design manual will give you an idea about what my design decisions are (what my intensions are), how the iPhone app will be used by presenting the model of the application which supplements this manual and explanations associated with them. This manual was created after several meetings and discussions with the Red Oak Financial Ltd that the project (iPhone app) based on their tax refund calculator and the technical details as the workings of the calculations cannot be published due to the Non-Disclosure Agreement with the company in this document.

The design of the modules in this manual are made in such a way as to separate the code of the GUI and the main code of the project that does the work so that other GUI's can be created in other environments at a later date with minimal changes needed to be made to the main program code. The GUI lay out in this manual is the user interfaces prototypes that assist in explanation of the project functionality.

2. Architectural Design



At the highest level, the software architecture above summarizes how the iPhone app will be organised in terms of sub-systems. The system is grouped into different layers (sub-systems) and each sub-system communicates with each other in a top-down fashion. The organisation of each sub-system is also further detailed via a modular design.

The Tax Refund Calculator iPhone app GUI layer is based on six modules. First four GUIs called Details GUI, Home GUI, Expenses GUI and Spouse GUI which are designated to get the relevant information (e.g. Annual Salary, Tax on Salary, Medical Expenses) from the user to calculate the tax refund estimation. One of the last two GUIs is called RefundResult GUI presents the tax refund estimation after the set of necessary calculations. Last GUI is called Apply GUI that allows the user to communicate with the company via email.

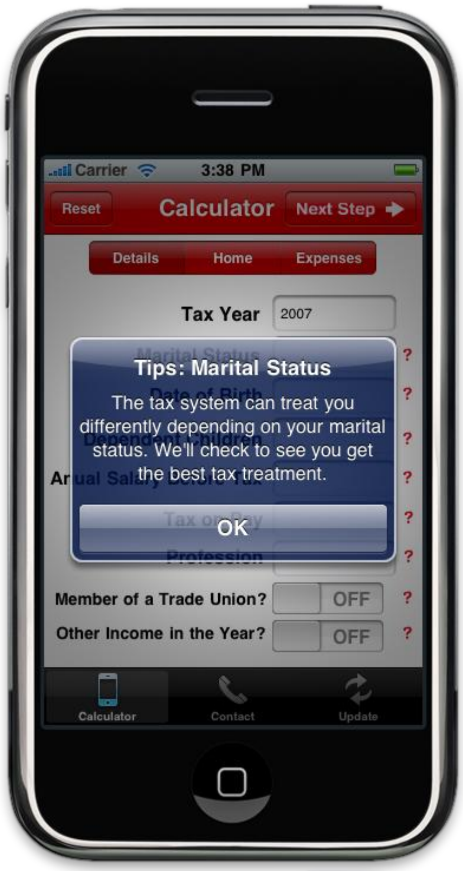
The Domain layer consists of nine modules. The classes DetailsViewController, HomeViewController, ExpensesViewController, RefundResultViewController and ApplyViewController that deal with the functionalities of the program interacted by the user in the relevant GUIs. The module called CalculateTaxRefund in this domain is very important since the set of calculations in order to get the refund estimation are made by this class. There is also module called ResetCalculator that cleans all the calculator fields which entered by the user and turns back to the calculator to its initial view (DetailsView). These modules are further detailed later section in this document.

Lastly, The services layer contains the class called Constants that imagine this class like a database holds all the constant values of the calculator that the calculator will be using these constants while performing the refund calculations.

3. Interfaces Design

Tax Refund Calculator has an explanatory help to clear any ambiguity of the user which associated with the each field of the calculator. The application will pop-up the blue message box which contains the relevant information, if the user taps the red question mark icon (?) beside the relevant field.

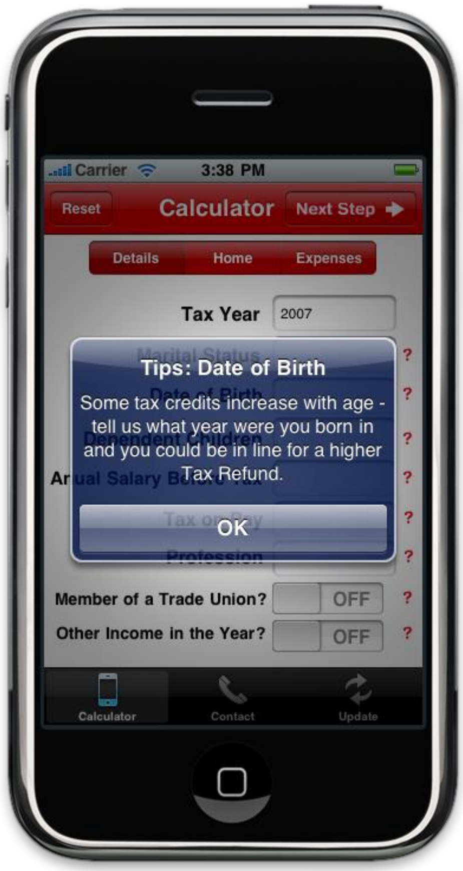
3.1. Help Icon (?) Tapped Screen Shots - “Details” Segmented Control



Tips: Marital Status

The tax system can treat you differently depending on your marital status. We'll check to see you get the best tax treatment.

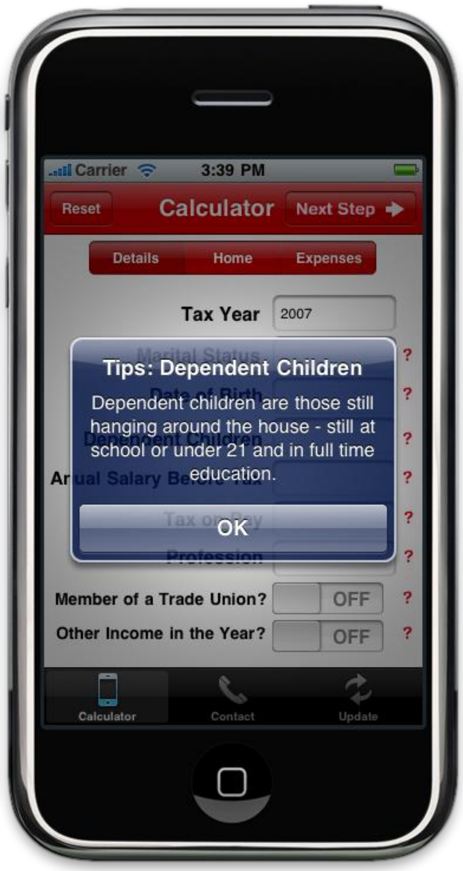
OK



Tips: Date of Birth

Some tax credits increase with age - tell us what year were you born in and you could be in line for a higher Tax Refund.

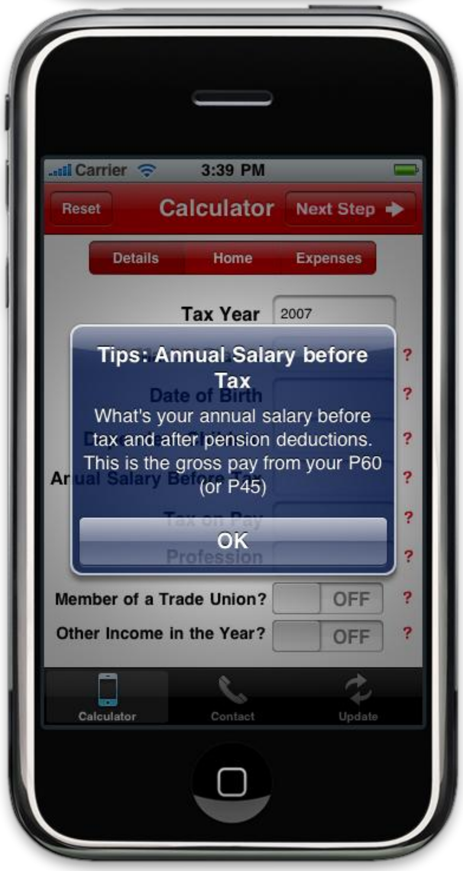
OK



Tips: Dependent Children

Dependent children are those still hanging around the house - still at school or under 21 and in full time education.

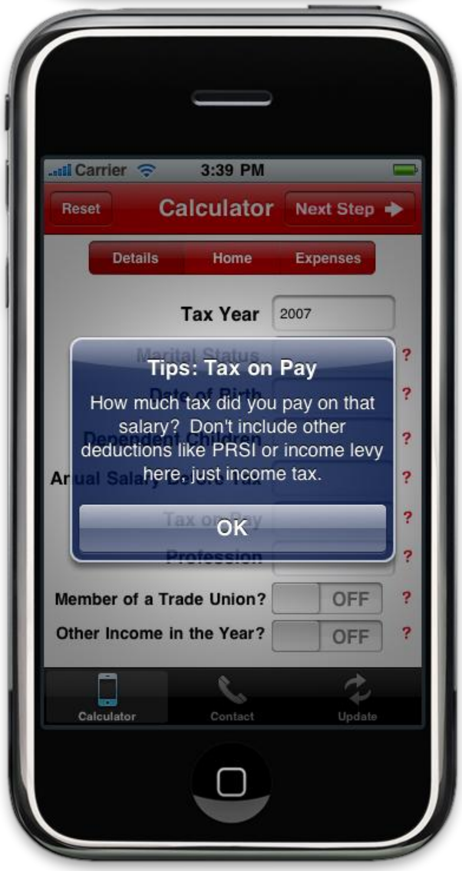
OK



Tips: Annual Salary before Tax

What's your annual salary before tax and after pension deductions. This is the gross pay from your P60 (or P45)

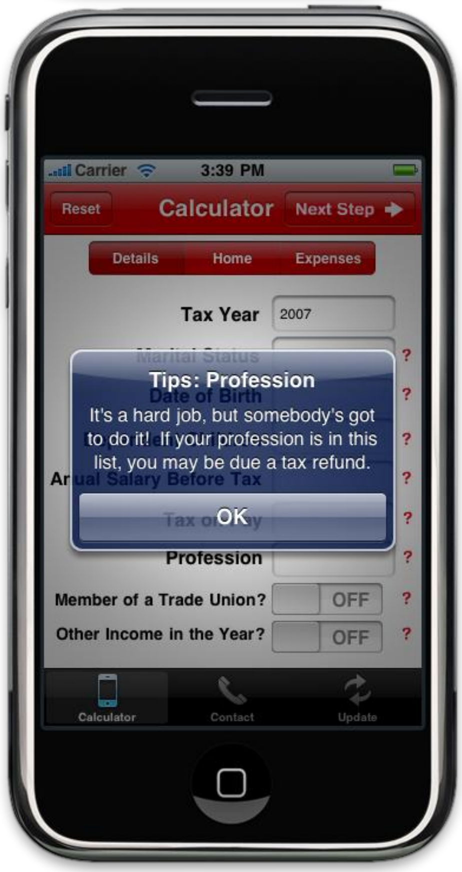
OK



Tips: Tax on Pay

How much tax did you pay on that salary? Don't include other deductions like PRSI or income levy here, just income tax.

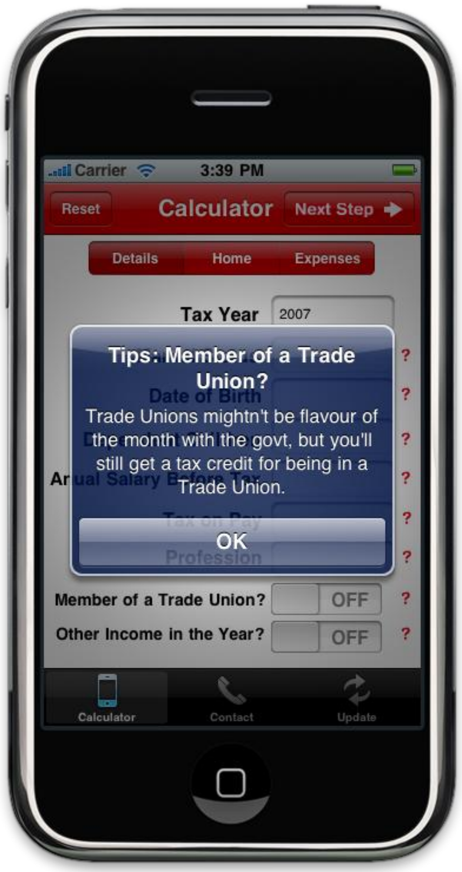
OK



Tips: Profession

It's a hard job, but somebody's got to do it! If your profession is in this list, you may be due a tax refund.

OK



Carrier 3:39 PM

Reset | Calculator | Next Step

Details | Home | Expenses

Tax Year: 2007

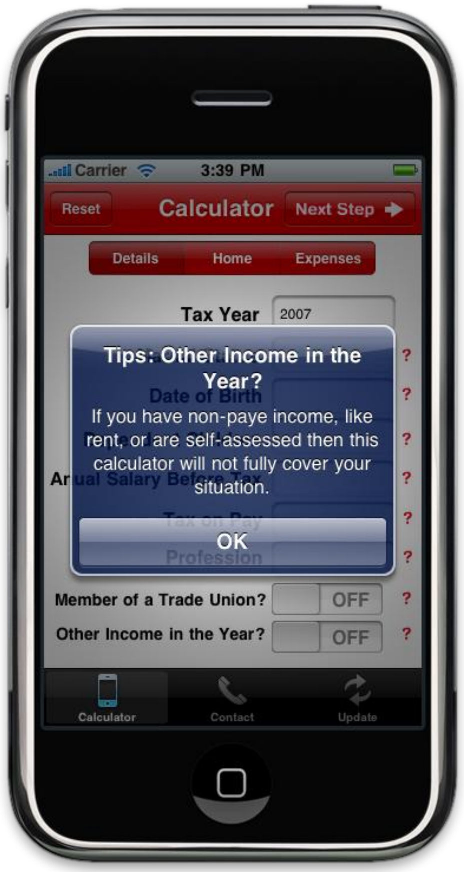
Tips: Member of a Trade Union?

Trade Unions mightn't be flavour of the month with the govt, but you'll still get a tax credit for being in a Trade Union.

OK

Member of a Trade Union? OFF

Other Income in the Year? OFF



Carrier 3:39 PM

Reset | Calculator | Next Step

Details | Home | Expenses

Tax Year: 2007

Tips: Other Income in the Year?

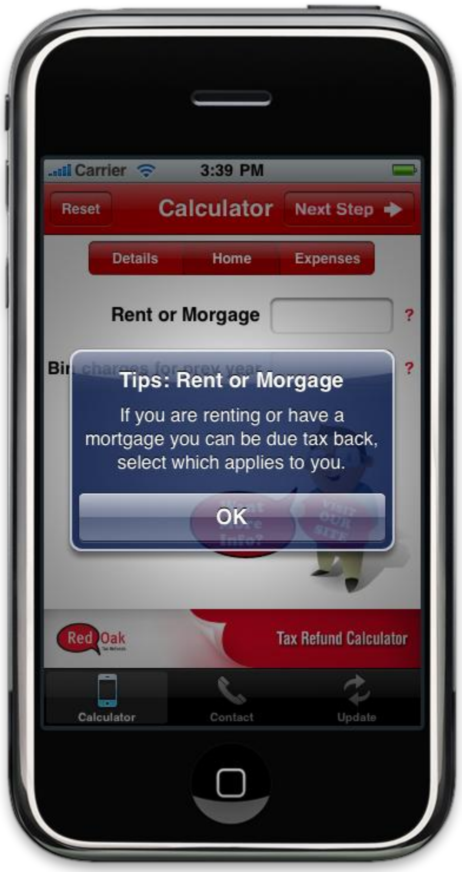
If you have non-paye income, like rent, or are self-assessed then this calculator will not fully cover your situation.

OK

Member of a Trade Union? OFF

Other Income in the Year? OFF

3.2. Help Icon (?) Tapped Screen Shots - "Home" Segmented Control



Carrier 3:39 PM

Reset | Calculator | Next Step

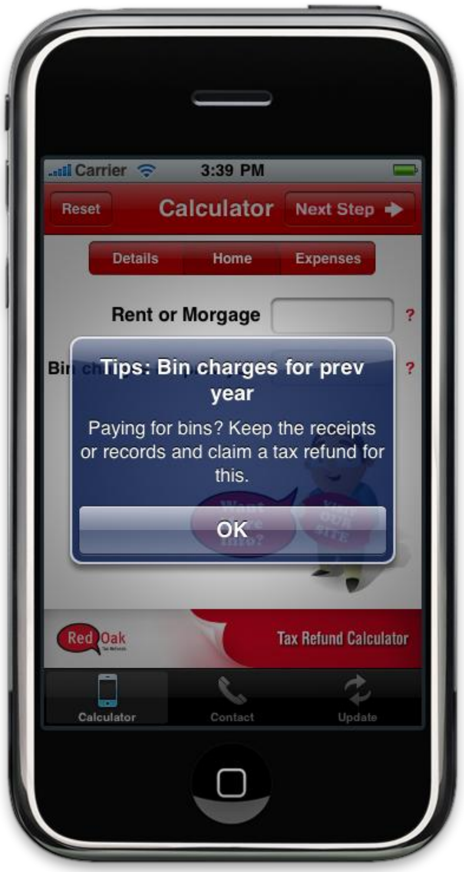
Details | Home | Expenses

Rent or Morgage

Tips: Rent or Morgage

If you are renting or have a mortgage you can be due tax back, select which applies to you.

OK



Carrier 3:39 PM

Reset | Calculator | Next Step

Details | Home | Expenses

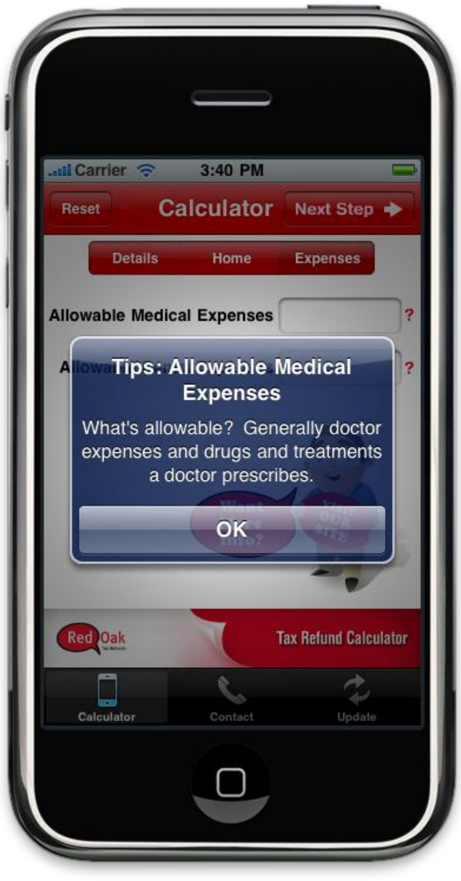
Rent or Morgage

Tips: Bin charges for prev year

Paying for bins? Keep the receipts or records and claim a tax refund for this.

OK

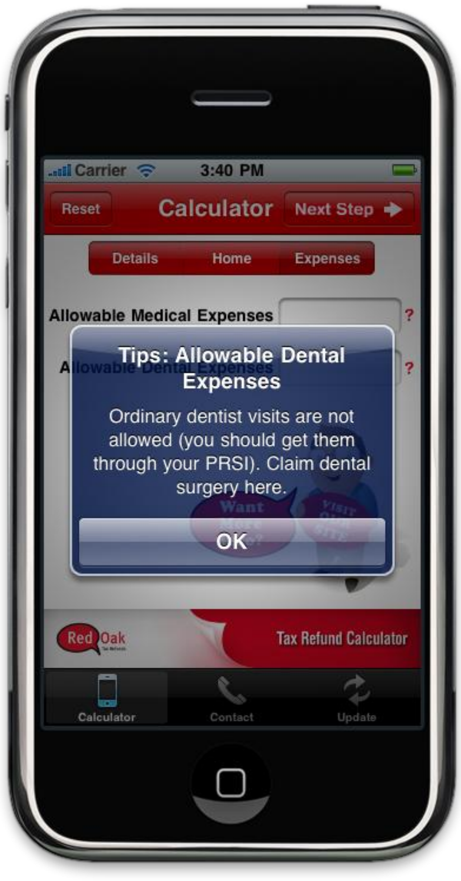
3.3. Help Icon (?) Tapped Screen Shots - “Expenses” Segmented Control



Tips: Allowable Medical Expenses

What's allowable? Generally doctor expenses and drugs and treatments a doctor prescribes.

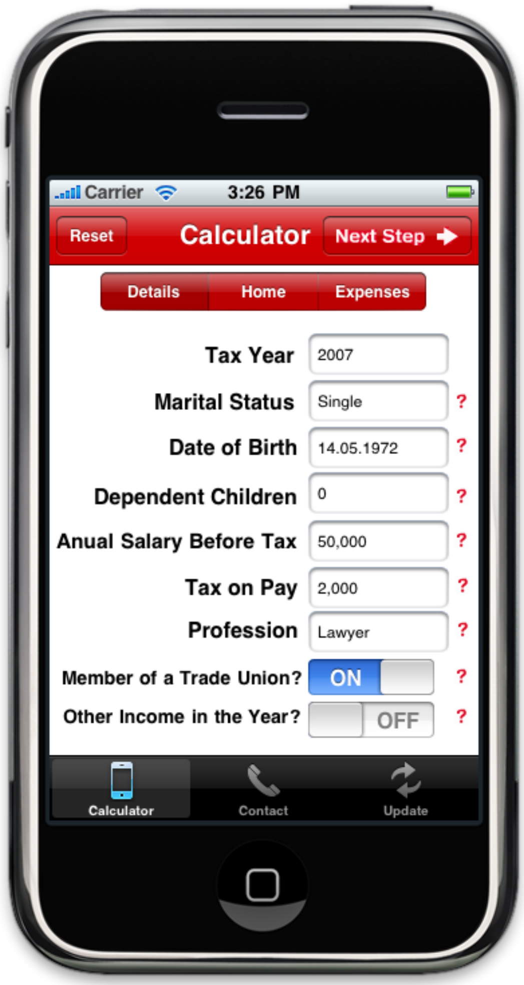
OK



Tips: Allowable Dental Expenses

Ordinary dentist visits are not allowed (you should get them through your PRSI). Claim dental surgery here.

OK


Carrier 3:26 PM

Reset **Calculator** Next Step

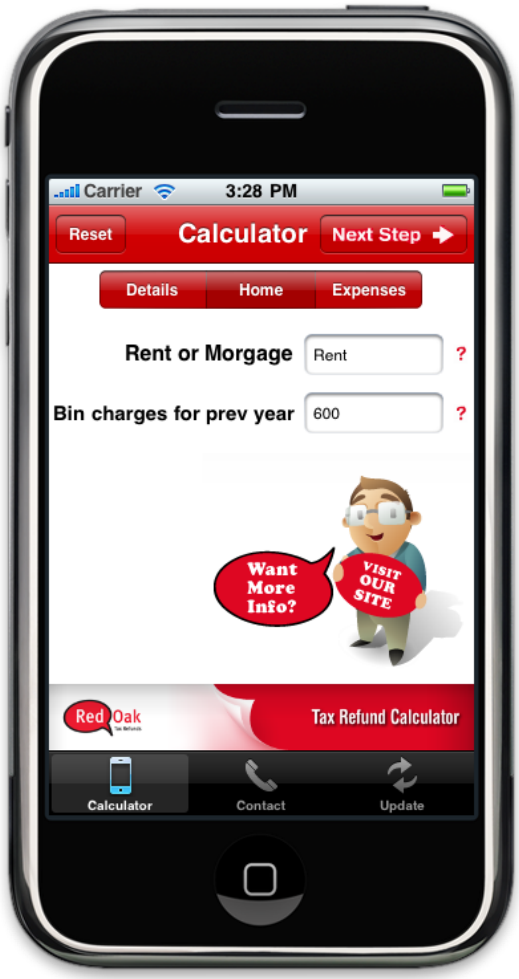
Details | Home | Expenses

Tax Year	2007
Marital Status	Single
Date of Birth	14.05.1972
Dependent Children	0
Anual Salary Before Tax	50,000
Tax on Pay	2,000
Profession	Lawyer
Member of a Trade Union?	ON
Other Income in the Year?	OFF

3.4. GUI Prototype Screen Shots

<u>"Details" View Screen Shot</u>

This screen shot is the start-up and the first step of the Tax Refund Calculator iPhone app called "Details" that allows the user enters his/her personal income tax information. The user can proceed next step "Home" either by touching the "Next Step" button above at the navigation bar or the user is able to immediate access to any step ("Details", "Home", or "Expenses") he/she wants by using the segmented control. "Reset" button is also provided on the navigation bar, if the user wishes to reload the calculator and clear all the fields contained by the "Details", "Home", "Expenses" and "Your Tax Refund" views.


Carrier 3:28 PM

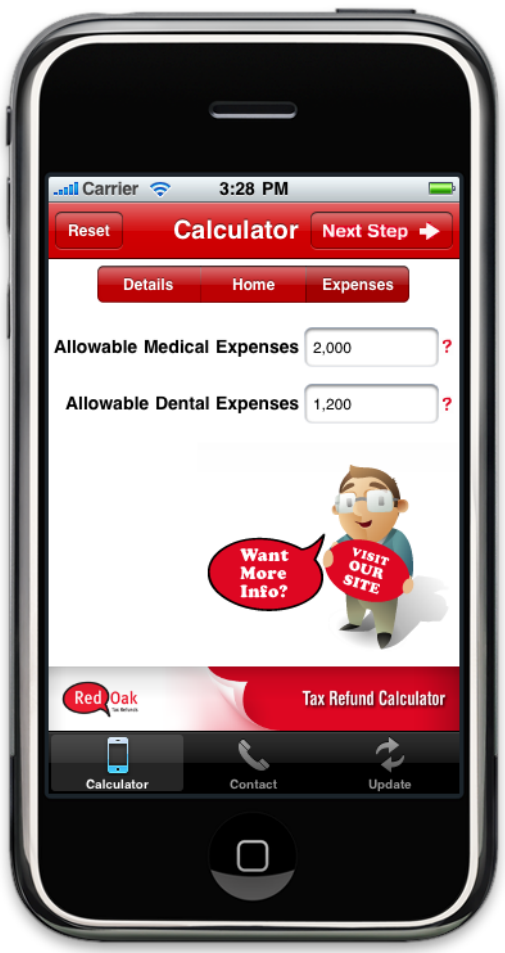
Reset **Calculator** Next Step

Details | Home | Expenses

Rent or Morgage	Rent
Bin charges for prev year	600

<u>"Home" View Screen Shot</u>

This screen shot is the second step of the Tax Refund Calculator iPhone app called "Home" that allows the user enters the information about his/her principle residence. The user can proceed next step "Expenses" either by touching the "Next Step" button above at the navigation bar or by using the segmented control. Reset button also exists on the navigation bar to reload the calculator.

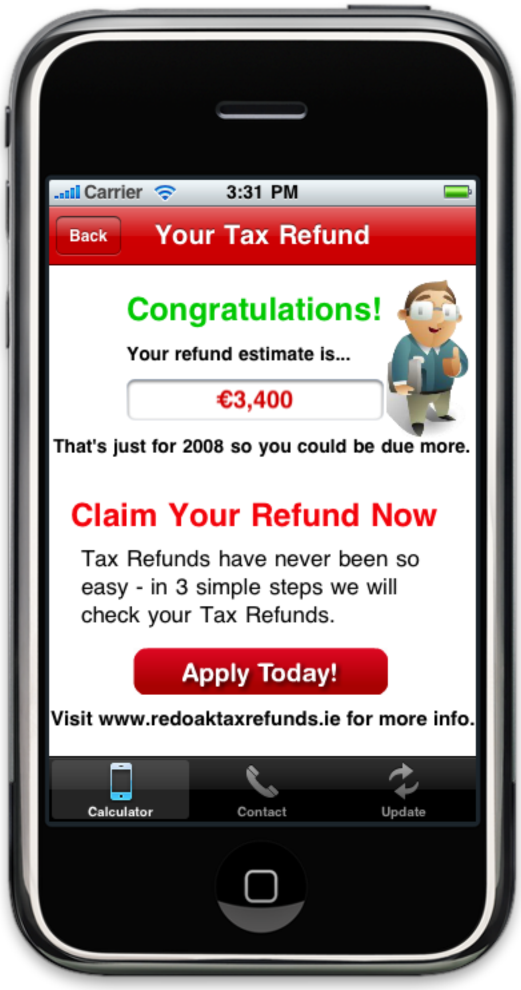


Allowable Medical Expenses 2,000

Allowable Dental Expenses 1,200

<u>“Expenses” View Screen Shot</u>

This screen shot is the third and the last step of the Tax Refund Calculator iPhone app called “Expenses” that allows the user enters the information about allowable expenses before presenting the result of the estimated tax refund. The user gets his/her tax refund estimation whenever he/she touches the “Next Step”, the calculator will switch and display the view “Your Tax Refund”. Reset button also exists on the navigation bar to reload the calculator.



Congratulations!

Your refund estimate is...

€3,400

That's just for 2008 so you could be due more.

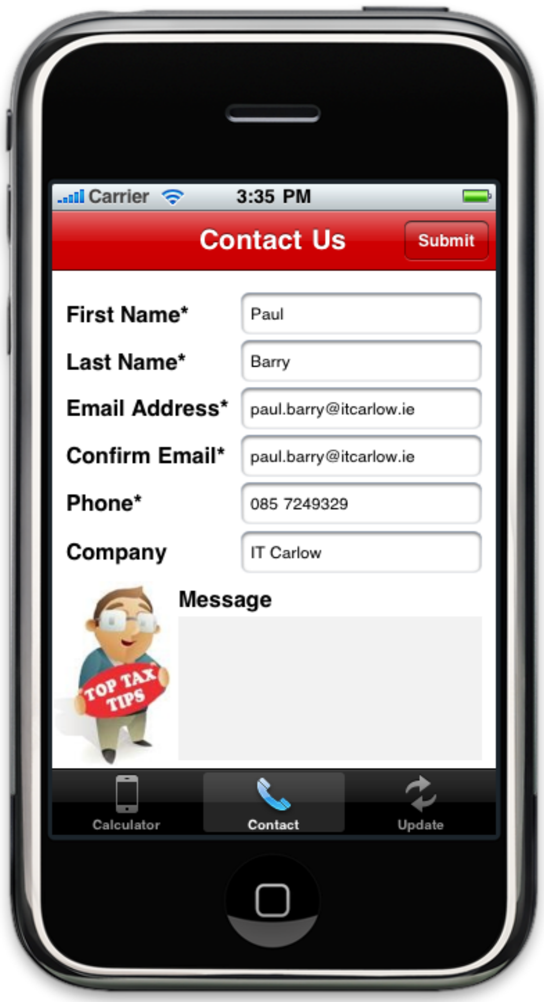
Claim Your Refund Now

Tax Refunds have never been so easy - in 3 simple steps we will check your Tax Refunds.

Visit www.redoaktaxrefunds.ie for more info.

<u>“Your Tax Refund” View Screen Shot</u>

This screen shot displays the estimated tax refunds of the user. The user is allowed to come back to the calculator (“Details” view) by touching the “Back” button on the navigation bar above, if the user doesn’t satisfy with the result and wants to make changes. The calculator dynamically calculates and fills the user refund estimation as long as the user fills the information without requiring further button for the calculation.



First Name*	Paul
Last Name*	Barry
Email Address*	paul.barry@itcarlow.ie
Confirm Email*	paul.barry@itcarlow.ie
Phone*	085 7249329
Company	IT Carlow

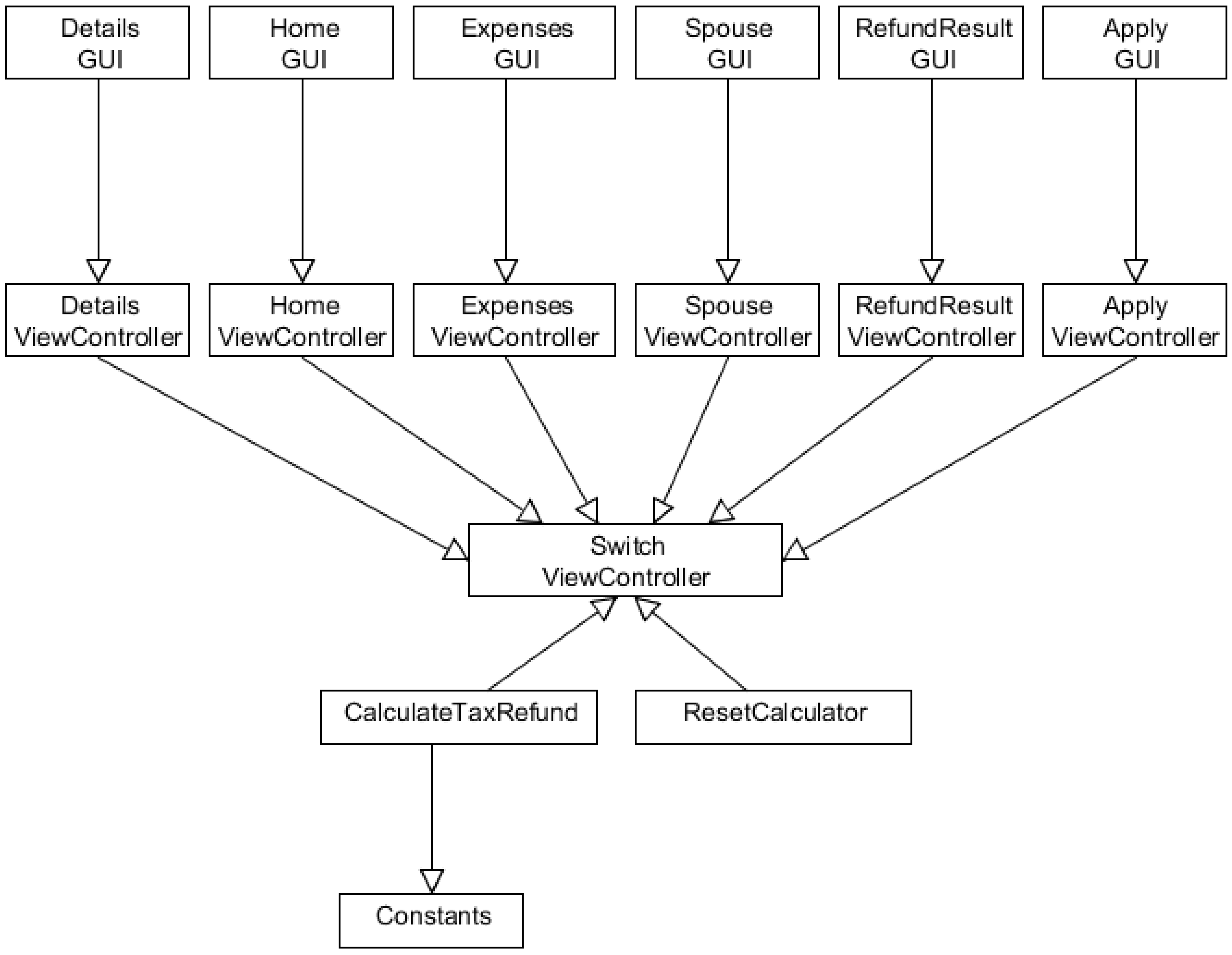
<u>"Contact us" View Screen Shot</u>

This screen shot allows the user to contact with the Red Oak Tax Refunds Company. The user enters the required fields and touches the "Submit" button in order to establish communication channel with the company. "Contact us" is last but not least view since this iPhone app's aim is to attract the users with the result of the calculation and makes them to apply Red Oak Tax Refunds Company to get back their tax refunds.

4. Modular Design of Sub-Systems

4.1. Overview

The diagram below is the visual representation of the iPhone app modular design.



4.2. Brief Responsibilities of Each Module

<u>DetailsViewController</u>

This module is a sub-class of the UIViewController which is a generic controller base class that manages a view. DetailsViewController hides all the information necessary for the functionalities of the Details view's elements such as when the user taps the date of birth picker button, DetailsViewController will respond this by sliding the date picker upwards on the screen or if the user has dependent children and he/she is Single, the module will respond this by making the "Are you cohabiting?" question visible.

HomeViewController

This module is a sub-class of the UIViewController which hides all the information necessary for the functionalities of the Home view's elements such as if the user taps the "Done" button on the keyboard, HomeViewController will resign the keyboard.

ExpensesViewController

This module is a sub-class of the UIViewController which hides all the information necessary for the functionalities of the Expenses view's elements such as if the user taps the Rent or Mortgage picker button and selects the Rent, ExpensesViewController will respond this by making the "Rent From" field visible.

The classes called SpouseViewController and RefundResultViewController will perform with the same fashion as in explained above modules.

ApplyViewController

This module is a sub-class of the UIViewController which hides all the information necessary for the functionalities of the Apply view's elements such as if the user taps the "Submit" button on the navigation bar, ApplyViewController will respond this by sliding the email composer picker to the screen. This module is also responsible to send an email (which contains the customer details) to the company.

SwitchViewController

This module is also a sub-class of the UIViewController which will be the very crucial class since this will be the head of the entire classes that manages the switches between the views and controlling the modules called CalculateTaxRefund and ResetCalculator. In other words, the overall mechanism of the calculator will be based on SwitchViewController.

CalculateTaxRefund

This module will get the user information from the SwitchViewController class and performs the necessary calculations in order to get the tax refund estimation of a user.

Reset Calculator

This module will access the user information from the SwitchViewController class and bring an application to initial state by clearing all the relevant fields in the calculator.

Constants

This module will hold all the relevant constant values for the calculator such as maximum Tax Rate of the year 2009 is 20% while minimum is 41%.

4.3. Data Structure Design of Major Modules

CalculateTaxRefund

This module data structure will be of an NSArray object which will be filled from the calculator fields through its indexes when the user taps a "Next Step" button on the navigation bar before getting his/her tax refund estimation. The module will access the each value of an array indexes to perform set of calculations.

ResetCalculator

This module data structure will also be of an NSArray object which will be filled from the calculator fields through its indexes when the user taps a "Reset" button on the navigation bar and the calculator will be turned back to its initial condition by the module which will set the relevant array indexes to the initial values.

ApplyViewController

This module will be using NSString object in order to get the user details when the user intends to send his/her details to the company. ApplyViewController class will perform operations to send an email with this NSString object as the body of an email.

4.4. Procedural Design of Major Modules

<u>SwitchViewController</u>

Starts when the user taps one of the segments (“Details” ,”Home” ,”Expenses” or ”Spouse”) of the Calculator Segmented Control in order to switch into another view.

1. SwitchViewController checks the selected segment index.
2. The appropriate functions are called to resign the picker view, date picker and keyboard, if they are active.
3. Removes the current view from being a superview without loosing any data that the user has already entered on to this view. (e.g. text field, picker view, slider)
4. Finally, inserts a selected segment’s view as a subview in order to switch into it.

Starts when the user taps the “Next Step” button on the navigation bar just before getting his/her tax refund estimation.

1. The appropriate functions are called to resign the picker view, date picker or keyboard, if they are active.
2. Removes the current view (“Expenses” or “Spouse”) from being a superview without loosing any data that the user has already entered on to this view.
3. Inserts “RefundResult” view as a subview by using RefundResultViewController class in order to switch into it.
4. Finally, calls the CalculateTaxRefund class which calculates the user refund estimation.

Starts when the user taps the “Reset” button on the navigation bar.

1. The appropriate functions are called to resign the picker view, date picker or keyboard, if they are active.
2. Removes the current view from being a superview and turns back to initial view “Details”.
3. Finally, calls the ResetCalculator class in order to reload the calculator.

Starts when the user taps the “Apply Today!” button on the “Refund Result” view.

1. Switch the tab bar controller into “Contact” tab bar item and displays the contact form.

Starts when the user taps the “Back” button on the navigation bar of the “RefundResult” view.

1. Removes the “RefundResult” view from being a superview and turns back to the view that the user was previously on.

<u>CalculateTaxRefund</u>

Starts when the user taps the “Next Step” button just before getting his/her tax refund estimation.

1. Gets the user details from the calculator.
2. Performs a set of calculations to determine the refund estimation of a user.
3. Passes the result to the appropriate view (“RefundResult” view).

ResetCalculator

Starts when the user taps the “Reset” button on the navigation bar.

1. Pop-up the message box for confirmation.
2. If the user confirms the operation, reload all the elements of each view to their initial values.
2. If the user cancels the operation, do nothing.

5. Test Provisions

The classes created during the project are being made with efficiency and ease of testing foremost in my mind. Testing the project as a whole should not be to difficult due to the fact that I should be able to test most of the modules individually. Testing will take place in the iPod Touch regularly straight after the completion of the major bunch of tasks.

6. Conclusion

It is an inevitable fact that design process is a very crucial step which should be done very carefully and properly before starting the coding of the project. After creating that file as you have seen above, now coding will be straight-forward and it will save time during programming of an iPhone application because of all the major design decisions will already have been taken. However, as soon as I begin writing code changes may be needed to the design document, it maybe some changes about architectural design, modular design or interfaces design. It is important to watch out also while coding, a high level of cohesion and a low level of coupling between modules is important to maintain. After that approach, it will be easy to test, debug and maintain.