

2. Classes

2.1. SwitchViewController.h

```
1. //
2. //  SwitchViewController.h
3. //  View_Switch
4. //
5. //  Created by C00115609 on 28/01/2010.
6. //  Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8.
9. #import <UIKit/UIKit.h>
10. #import "DetailsViewController.h"
11.
12. @class ApplyViewController;
13. @class RefundResultViewController;
14. @class SpouseViewController;
15. @class ExpensesViewController;
16. @class MyTaxViewController;
17. @class DetailsViewController;
18. @class CalculateTaxRefund;
19. @class ResetCalculator;
20.
21. @interface SwitchViewController : UIViewController <UITableViewDelegate,
    UITableViewDelegate >{
22.
23.     ApplyViewController *applyViewController;
24.     RefundResultViewController *refundResultViewController;
25.     SpouseViewController *spouseViewController;
26.     ExpensesViewController *expensesViewController;
27.     MyTaxViewController *myTaxViewController;
28.     DetailsViewController *detailsViewController;
29.
30.     CalculateTaxRefund *calculateTaxRefund;
31.     ResetCalculator *resetCalculator;
```

```
32.
33.     IBOutlet UISegmentedControl *calculatorSegment;
34.     IBOutlet UINavigationController *titleNavigationBar;
35.     IBOutlet UINavigationController *refundResultBar;
36.     IBOutlet UIButton *applyTodayButton;
37.     IBOutlet UITabBarController *rootController;
38. }
39.
40. @property (nonatomic, retain) ApplyViewController *applyViewController;
41. @property (nonatomic, retain) RefundResultViewController *refundResultViewController;
42. @property (nonatomic, retain) SpouseViewController *spouseViewController;
43. @property (nonatomic, retain) ExpensesViewController *expensesViewController;
44. @property (nonatomic, retain) MyTaxViewController *myTaxViewController;
45. @property (nonatomic, retain) DetailsViewController *detailsViewController;
46.
47. @property (nonatomic, retain) CalculateTaxRefund *calculateTaxRefund;
48. @property (nonatomic, retain) ResetCalculator *resetCalculator;
49. @property (nonatomic, retain) IBOutlet UITabBarController *rootController;
50. @property (nonatomic, retain) IBOutlet UISegmentedControl *calculatorSegment;
51.
52. - (IBAction) calculateBarButtonPressed:(id) sender;
53. - (IBAction) resetButtonPressed:(id) sender;
54. - (IBAction) switchSegmentViews;
55. - (IBAction) backButtonPressed:(id) sender;
56. - (IBAction) applyTodayButtonPressed:(id) sender;
57.
58. - (void) switchToRefundResult;
59. - (void) switchToSpouse;
60. - (void) switchToExpenses;
61. - (void) switchToMyTax;
62. - (void) switchToDetails;
63. - (void) resignPickerView;
64. - (void) resignDatePicker;
65. - (void) resignKeyboard;
66.
67. - (void) maritalStatus: (DetailsViewController *) controller;
```

```
68. - (void) dialogOtherAction;
69.
70. @end
```

2.2. SwitchViewController.m

```
1. //
2. // SwitchViewController.m
3. // View_Switch
4. //
5. // Created by C00115609 on 28/01/2010.
6. // Copyright 2010 __Tuna Erduramaz__. All rights reserved.
7. //
8. /*****
9.             Class Description
10. This class is responsible for controlling entire view's of the application and it also contains
11. the main navigation bar functionalities.
12. *****/
13.
14. #import "SwitchViewController.h"
15. #import "ApplyViewController.h"
16. #import "RefundResultViewController.h"
17. #import "SpouseViewController.h"
18. #import "ExpensesViewController.h"
19. #import "MyTaxViewController.h"
20. #import "DetailsViewController.h"
21. #import "CalculateTaxRefund.h"
22. #import "ResetCalculator.h"
23.
24. @implementation SwitchViewController
25.
26. @synthesize applyViewController;
27. @synthesize refundResultViewController;
28. @synthesize spouseViewController;
```

```

29. @synthesize expensesViewController;
30. @synthesize myTaxViewController;
31. @synthesize detailsViewController;
32. @synthesize rootController;
33. @synthesize calculatorSegment;
34. @synthesize calculateTaxRefund, resetCalculator;
35.
36. // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
37. - (void)viewDidLoad {
38.     [super viewDidLoad];
39.
40.     DetailsViewController *detailsController = [[DetailsViewController alloc]
41.                                                initWithNibName:@"DetailsView"
bundle:nil];
42.     self.detailsViewController = detailsController;
43.     [self.view addSubview:detailsController.view atIndex:0];
44.     [detailsController release];
45.
46.     [detailsController setDelegate:self];
47.
48.     /* Make the segment's views invisible interfaces visible */
49.     titleNavigationBar.hidden=FALSE;
50.     calculatorSegment.hidden=FALSE;
51.
52.     CalculateTaxRefund *calculateRefund = [[CalculateTaxRefund alloc] init];
53.     self.calculateTaxRefund = calculateRefund;
54.
55.     ResetCalculator *resetCalc = [[ResetCalculator alloc] init];
56.     self.resetCalculator = resetCalc;
57. }
58.
59. - (void) switchToRefundResult{
60.     [UIView beginAnimations:@"View Flip" context:nil];
61.     [UIView setAnimationDuration:1.0];
62.     [UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];
63.

```

```

64.     if (self.refundResultViewController == nil)
65.     {
66.         RefundResultViewController *refundResulController =
67.         [[RefundResultViewController alloc] initWithNibName:@"RefundResultView"
68.         bundle:nil];
69.         self.refundResultViewController = refundResulController;
70.         [refundResulController release];
71.     }
72.
73.     [UIView setAnimationTransition:
74.     UIViewAnimationTransitionCurlDown
75.     //UIViewAnimationTransitionFlipFromLeft
76.     forView:self.view cache:YES];
77.     /* Switch the view to the Refund Result */
78.     if (self.detailsViewController.view.superview != nil){
79.         [refundResultViewController viewWillAppear:YES];
80.         [detailsViewController viewWillDisappear:YES];
81.         [detailsViewController.view removeFromSuperview];
82.         [self.view insertSubview:refundResultViewController.view atIndex:0];
83.         [detailsViewController viewDidDisappear:YES];
84.         [refundResultViewController viewDidAppear:YES];
85.     }
86.     else if (self.myTaxViewController.view.superview != nil){
87.         [refundResultViewController viewWillAppear:YES];
88.         [myTaxViewController viewWillDisappear:YES];
89.         [myTaxViewController.view removeFromSuperview];
90.         [self.view insertSubview:refundResultViewController.view atIndex:0];
91.         [myTaxViewController viewDidDisappear:YES];
92.         [refundResultViewController viewDidAppear:YES];
93.     }
94.     else if (self.expensesViewController.view.superview != nil){
95.         [refundResultViewController viewWillAppear:YES];
96.         [expensesViewController viewWillDisappear:YES];
97.         [expensesViewController.view removeFromSuperview];
98.         [self.view insertSubview:refundResultViewController.view atIndex:0];

```

```

99.         [expensesViewController viewDidDisappear:YES];
100.        [refundResultViewController viewDidAppear:YES];
101.    }
102.    else if (self.spouseViewController.view.superview != nil){
103.        [refundResultViewController viewWillAppear:YES];
104.        [spouseViewController viewWillDisappear:YES];
105.        [spouseViewController.view removeFromSuperview];
106.        [self.view insertSubview:refundResultViewController.view atIndex:0];
107.        [spouseViewController viewDidDisappear:YES];
108.        [refundResultViewController viewDidAppear:YES];
109.    }
110.
111.    [UIView commitAnimations];
112.
113.    /* Make the Refund Result's view invisible interface visible and
114.       the segment's views visible interfaces invisible */
115.    titleNavigationBar.hidden=TRUE;
116.    calculatorSegment.hidden=TRUE;
117.    refundResultBar.hidden=FALSE;
118.    applyTodayButton.hidden=FALSE;
119. }
120.
121. - (void) switchToSpouse{
122.     if (self.spouseViewController == nil)
123.     {
124.         SpouseViewController *spouseController =
125.         [[SpouseViewController alloc] initWithNibName:@"SpouseView"
126.                                                    bundle:nil];
127.         self.spouseViewController = spouseController;
128.         [spouseController release];
129.     }
130.
131.     /* Switch the view to the Spouse */
132.     if (self.detailsViewController.view.superview != nil){
133.         [spouseViewController viewWillAppear:YES];
134.         [detailsViewController viewWillDisappear:YES];

```

```

135.         [detailsViewController.view removeFromSuperview];
136.         [self.view insertSubview:spouseViewController.view atIndex:0];
137.         [detailsViewController viewDidDisappear:YES];
138.         [spouseViewController viewDidAppear:YES];
139.     }
140.     else if (self.myTaxViewController.view.superview != nil){
141.         [spouseViewController viewWillAppear:YES];
142.         [myTaxViewController viewWillDisappear:YES];
143.         [myTaxViewController.view removeFromSuperview];
144.         [self.view insertSubview:spouseViewController.view atIndex:0];
145.         [myTaxViewController viewDidDisappear:YES];
146.         [spouseViewController viewDidAppear:YES];
147.     }
148.     else if (self.expensesViewController.view.superview != nil){
149.         [spouseViewController viewWillAppear:YES];
150.         [expensesViewController viewWillDisappear:YES];
151.         [expensesViewController.view removeFromSuperview];
152.         [self.view insertSubview:spouseViewController.view atIndex:0];
153.         [expensesViewController viewDidDisappear:YES];
154.         [spouseViewController viewDidAppear:YES];
155.     }
156.     /* Set new next step button position */
157.     //nextStepButton.frame = CGRectMake(180, 250, 123, 23);
158. }
159.
160. - (void) switchToExpenses{
161.     if (self.expensesViewController == nil)
162.     {
163.         ExpensesViewController *expensesController =
164.         [[ExpensesViewController alloc] initWithNibName:@"ExpensesView"
165.                                                     bundle:nil];
166.         self.expensesViewController = expensesController;
167.         [expensesController release];
168.     }
169.
170.     /* Switch the view to the Expenses */

```

```

171.     if (self.detailsViewController.view.superview != nil){
172.         [expensesViewController viewWillAppear:YES];
173.         [detailsViewController viewWillDisappear:YES];
174.         [detailsViewController.view removeFromSuperview];
175.         [self.view insertSubview:expensesViewController.view atIndex:0];
176.         [detailsViewController viewDidDisappear:YES];
177.         [expensesViewController viewDidAppear:YES];
178.     }
179.     else if (self.myTaxViewController.view.superview != nil){
180.         [expensesViewController viewWillAppear:YES];
181.         [myTaxViewController viewWillDisappear:YES];
182.         [myTaxViewController.view removeFromSuperview];
183.         [self.view insertSubview:expensesViewController.view atIndex:0];
184.         [myTaxViewController viewDidDisappear:YES];
185.         [expensesViewController viewDidAppear:YES];
186.     }
187.     else if (self.spouseViewController.view.superview != nil){
188.         [expensesViewController viewWillAppear:YES];
189.         [spouseViewController viewWillDisappear:YES];
190.         [spouseViewController.view removeFromSuperview];
191.         [self.view insertSubview:expensesViewController.view atIndex:0];
192.         [spouseViewController viewDidDisappear:YES];
193.         [expensesViewController viewDidAppear:YES];
194.     }
195. }
196.
197. - (void) switchToMyTax{
198.     if (self.myTaxViewController == nil)
199.     {
200.         MyTaxViewController *myTaxController =
201.         [[MyTaxViewController alloc] initWithNibName:@"MyTaxView"
202.                                                bundle:nil];
203.         self.myTaxViewController = myTaxController;
204.         [myTaxController release];
205.     }
206.

```



```

207.     /* Switch the view to the MyTax */
208.     if (self.detailsViewController.view.superview != nil){
209.         [myTaxViewController viewWillAppear:YES];
210.         [detailsViewController viewWillDisappear:YES];
211.         [detailsViewController.view removeFromSuperview];
212.         [self.view insertSubview:myTaxViewController.view atIndex:0];
213.         [detailsViewController viewDidDisappear:YES];
214.         [myTaxViewController viewDidAppear:YES];
215.     }
216.     else if (self.expensesViewController.view.superview != nil){
217.         [myTaxViewController viewWillAppear:YES];
218.         [expensesViewController viewWillDisappear:YES];
219.         [expensesViewController.view removeFromSuperview];
220.         [self.view insertSubview:myTaxViewController.view atIndex:0];
221.         [expensesViewController viewDidDisappear:YES];
222.         [myTaxViewController viewDidAppear:YES];
223.     }
224.     else if (self.spouseViewController.view.superview != nil){
225.         [myTaxViewController viewWillAppear:YES];
226.         [spouseViewController viewWillDisappear:YES];
227.         [spouseViewController.view removeFromSuperview];
228.         [self.view insertSubview:myTaxViewController.view atIndex:0];
229.         [spouseViewController viewDidDisappear:YES];
230.         [myTaxViewController viewDidAppear:YES];
231.     }
232. }
233.
234. - (void) switchToDetails{
235.
236.     /* Switch the view to the Details */
237.     if (self.myTaxViewController.view.superview != nil){
238.         [detailsViewController viewWillAppear:YES];
239.         [myTaxViewController viewWillDisappear:YES];
240.         [myTaxViewController.view removeFromSuperview];
241.         [self.view insertSubview:detailsViewController.view atIndex:0];
242.         [myTaxViewController viewDidDisappear:YES];

```

```

243.         [detailsViewController viewDidAppear:YES];
244.     }
245.     else if (self.expensesViewController.view.superview != nil){
246.         [detailsViewController viewWillAppear:YES];
247.         [expensesViewController viewWillDisappear:YES];
248.         [expensesViewController.view removeFromSuperview];
249.         [self.view insertSubview:detailsViewController.view atIndex:0];
250.         [expensesViewController viewDidDisappear:YES];
251.         [detailsViewController viewDidAppear:YES];
252.     }
253.     else if (self.spouseViewController.view.superview != nil){
254.         [detailsViewController viewWillAppear:YES];
255.         [spouseViewController viewWillDisappear:YES];
256.         [spouseViewController.view removeFromSuperview];
257.         [self.view insertSubview:detailsViewController.view atIndex:0];
258.         [spouseViewController viewDidDisappear:YES];
259.         [detailsViewController viewDidAppear:YES];
260.     }
261. }
262.
263. #pragma mark -
264. #pragma mark DetailsViewDelegate
265. //Custom protocol from DetailsViewController to listen IBAction
266. -(void) maritalStatus: (DetailsViewController *) controller {
267.     // Insert new segment with title "Spouse"
268.     if ([[[[detailsViewController maritalStatusPickerButton] titleLabel] text]
269.         isEqualToString:@"Married"]) {
270.         calculatorSegment.frame = CGRectMake(16, 54, 287, 30);
271.         [calculatorSegment insertSegmentWithTitle: @"4) Spouse" atIndex: 3 animated: YES];
272.     }
273.     // Remove segment with title "Spouse"
274.     else {
275.         calculatorSegment.frame = CGRectMake(32, 54, 257, 30);
276.         [calculatorSegment removeSegmentAtIndex:3 animated:YES];
277.     }

```

```
278.
279. /* Turn back to calculator into previously active segment */
280. -(IBAction) backButtonPressed:(id)sender{
281.
282.     [UIView beginAnimations:@"View Flip" context:nil];
283.         [UIView setAnimationDuration:1.00];
284.         [UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];
285.
286.     if (self.refundResultViewController.view.superview != nil){
287.         if([calculatorSegment selectedSegmentIndex] == 0){
288.             [detailsViewController viewWillAppear:YES];
289.             [refundResultViewController viewWillDisappear:YES];
290.             [refundResultViewController.view removeFromSuperview];
291.             [self.view insertSubview:detailsViewController.view atIndex:0];
292.             [refundResultViewController viewDidDisappear:YES];
293.             [detailsViewController viewDidAppear:YES];
294.             calculatorSegment.selectedSegmentIndex = 0;
295.         }
296.         if([calculatorSegment selectedSegmentIndex] == 1){
297.             [myTaxViewController viewWillAppear:YES];
298.             [refundResultViewController viewWillDisappear:YES];
299.             [refundResultViewController.view removeFromSuperview];
300.             [self.view insertSubview:myTaxViewController.view atIndex:0];
301.             [refundResultViewController viewDidDisappear:YES];
302.             [myTaxViewController viewDidAppear:YES];
303.             calculatorSegment.selectedSegmentIndex = 1;
304.         }
305.         else if([calculatorSegment selectedSegmentIndex] == 2){
306.             [expensesViewController viewWillAppear:YES];
307.             [refundResultViewController viewWillDisappear:YES];
308.             [refundResultViewController.view removeFromSuperview];
309.             [self.view insertSubview:expensesViewController.view atIndex:0];
310.             [refundResultViewController viewDidDisappear:YES];
311.             [expensesViewController viewDidAppear:YES];
312.             calculatorSegment.selectedSegmentIndex = 2;
313.         }

```

```

314.         else if([calculatorSegment selectedSegmentIndex] == 3){
315.             [spouseViewController viewWillAppear:YES];
316.             [refundResultViewController viewWillDisappear:YES];
317.             [refundResultViewController.view removeFromSuperview];
318.             [self.view insertSubview:spouseViewController.view atIndex:0];
319.             [refundResultViewController viewDidDisappear:YES];
320.             [spouseViewController viewDidAppear:YES];
321.             calculatorSegment.selectedSegmentIndex = 3;
322.         }
323.     }
324.
325.     [UIView setAnimationTransition:
326.         UIViewAnimationTransitionCurlUp
327.         forView:self.view cache:YES];
328.
329.     [UIView commitAnimations];
330.
331.     /* Make the Details' view invisible interface visible and
332.        the Refund Result's view visible interfaces invisible */
333.     titleNavigationBar.hidden=FALSE;
334.     calculatorSegment.hidden=FALSE;
335.     refundResultBar.hidden=TRUE;
336.     applyTodayButton.hidden=TRUE;
337. }
338.
339. /* Switch the view into Contact form */
340. -(IBAction) applyTodayButtonPressed:(id) sender{
341.
342.     if (self.applyViewController == nil)
343.     {
344.         ApplyViewController *applyController =
345.             [[ApplyViewController alloc] initWithNibName:@"ApplyView"
346.                                                     bundle:nil];
347.         self.applyViewController = applyController;
348.         [applyController release];
349.     }

```

```
350.
351.     /* Switch the contact tab */
352.     if (self.refundResultViewController.view.superview != nil){
353.         rootController.selectedIndex = 1;
354.     }
355. }
356.
357. -(IBAction)calculateBarButtonPressed:(id) sender{
358.     [self resignPickerView];
359.     [self resignDatePicker];
360.     [self resignKeyboard];
361.     [self switchToRefundResult];
362.
363.     [calculateTaxRefund taxRefundCalculation:self];
364. }
365.
366. -(IBAction) switchSegmentViews
367. {
368.     switch (calculatorSegment.selectedSegmentIndex) {
369.         case 0:
370.             [self resignPickerView];
371.             [self resignDatePicker];
372.             [self resignKeyboard];
373.             [self switchToDetails];
374.             break;
375.         case 1:
376.             [self resignPickerView];
377.             [self resignDatePicker];
378.             [self resignKeyboard];
379.             [self switchToMyTax];
380.             break;
381.         case 2:
382.             [self resignPickerView];
383.             [self resignDatePicker];
384.             [self resignKeyboard];
385.             [self switchToExpenses];
```

```
386.             break;
387.         case 3:
388.             [self resignPickerView];
389.             [self resignDatePicker];
390.             [self resignKeyboard];
391.             [self switchToSpouse];
392.             break;
393.         }
394.     }
395.
396. - (void) resignDatePicker{
397.     // If picker view is active then resign it
398.     if (detailsViewController.datePicker.superview != nil) {
399.         [detailsViewController resignDatePicker];
400.     }
401. }
402.
403. - (void) resignPickerView{
404.     // If picker view is active then resign it
405.     if(detailsViewController.pickerViewDetails.superview != nil){
406.         [detailsViewController resignPickerView];
407.     }
408.     if(expensesViewController.pickerViewRentOrMortgage.superview != nil){
409.         [expensesViewController resignPickerViewRentOrMortgage];
410.     }
411.     else if(expensesViewController.pickerViewRentFrom.superview != nil){
412.         [expensesViewController resignPickerViewRentFrom];
413.     }
414.     else if (spouseViewController.datePicker.superview != nil) {
415.         [spouseViewController resignPickerView];
416.     }
417. }
418.
419. - (void) resignKeyboard {
420.     [expensesViewController resignKeyboard];
421. }
```

```
422.
423. // Clear all the calculator fields
424. - (IBAction) resetButtonPressed:(id)sender
425. {
426.     // Pop up the dialog box for 'Reset' confirmation
427.     [self dialogOtherAction];
428. }
429.
430. // Thousand Separator
431. -(NSString*) commaSeparateThousands: (int) number
432. {
433.     NSString *result = @"";
434.     while( number > 0 )
435.     {
436.         int removedDigits = number % 1000;
437.         if( result != nil && [result length] > 0 )
438.         {
439.             if( number > 999 )
440.                 result = [NSString stringWithFormat:@"%%.3i,%@", removedDigits, result];
441.             else
442.                 result = [NSString stringWithFormat:@"%i,%@", removedDigits, result];
443.         }
444.         else
445.         {
446.             if( number > 999 )
447.                 result = [NSString stringWithFormat:@"%%.3i", removedDigits];
448.             else
449.                 result = [NSString stringWithFormat:@"%i", removedDigits];
450.         }
451.         number = number/1000;
452.     }
453.
454.     return result;
455. }
456.
457. // Dialog box for the 'Reset' button
```

```

458. - (void)dialogOtherAction
459. {
460.     // open a dialog with two custom buttons (Are you sure you want to reset the calculator?)
461.     UIAlertController *actionSheet = [[UIAlertSheet alloc] initWithTitle:@" "
462.     delegate:self cancelButtonTitle:nil destructiveButtonTitle:nil
463.     otherButtonTitles:@"Reset Calculator", @"Cancel", nil];
464.     actionSheet.actionSheetStyle = UIAlertControllerStyleDefault;
465.     actionSheet.destructiveButtonIndex = 0; // make the second button red (destructive)
466.     actionSheet.cancelButtonIndex = 1; // make the second button red (destructive)
467.     [actionSheet showInView:rootController.view]; // show from our table view (pops up in the
middle of the table)
468.     [actionSheet release];
469. }
470.
471. #pragma mark -
472. #pragma mark - UIAlertControllerDelegate
473.
474. - (void)actionSheet:(UIAlertSheet *)actionSheet clickedButtonAtIndex:(NSInteger)buttonIndex
475. {
476.     // the user clicked one of the OK/Cancel buttons
477.     if (buttonIndex == 0)
478.     {
479.         // Resign the active picker
480.         if(detailsViewController.pickerViewDetails.superview != nil){
481.             [self resignPickerView];
482.         }
483.         else if ((detailsViewController.datePicker.superview != nil)) {
484.             [self resignDatePicker];
485.         }
486.
487.         // Switch to view to DetailsView
488.         calculatorSegment.selectedSegmentIndex = 0;
489.
490.         // Reset the calculator

```



```
491.         [resetCalculator resetCalculatorInfo:self];
492.         // NSLog(@"ok");
493.     }
494.     else
495.     {
496.         // NSLog(@"cancel");
497.     }
498. }
499.
500. /*
501. // The designated initializer. Override if you create the controller programmatically and want to
    perform customization that is not appropriate for viewDidLoad.
502. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
503.     if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
504.         // Custom initialization
505.     }
506.     return self;
507. }
508. */
509.
510. /*
511. // Implement loadView to create a view hierarchy programmatically, without using a nib.
512. - (void)loadView {
513. }
514. */
515.
516. /*
517. // Override to allow orientations other than the default portrait orientation.
518. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
519.     // Return YES for supported orientations
520.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
521. }
522. */
523.
524. - (void)didReceiveMemoryWarning {
525.     // Releases the view if it doesn't have a superview.
```

```
526.     [super didReceiveMemoryWarning];
527.
528.     // Release any cached data, images, etc that aren't in use.
529.     // Release anything that's not essential, such as cached data
530. if (self.detailsViewController.view.superview == nil)
531.     self.detailsViewController = nil;
532. else if (self.myTaxViewController.view.superview == nil)
533.     self.myTaxViewController = nil;
534.     else if (self.expensesViewController.view.superview == nil)
535.         self.expensesViewController = nil;
536.     else if (self.refundResultViewController.view.superview == nil)
537.         self.refundResultViewController = nil;
538.     else if (self.applyViewController.view.superview == nil)
539.         self.applyViewController = nil;
540.     else if (self.spouseViewController.view.superview == nil)
541.         self.spouseViewController = nil;
542. }
543.
544. - (void)viewDidUnload {
545.     [super viewDidUnload];
546.
547.     // Release any retained subviews of the main view.
548.     // e.g. self.myOutlet = nil;
549. }
550.
551.
552. - (void)dealloc {
553.     [applyViewController release];
554.     [refundResultViewController release];
555.     [spouseViewController release];
556.     [expensesViewController release];
557.     [myTaxViewController release];
558.     [detailsViewController release];
559.     [rootController release];
560.     [calculatorSegment release];
561.     [calculateTaxRefund release];
```

```
562.         [resetCalculator release];
563.         [super dealloc];
564.     }
565.
566. @end
```

2.3. DetailsViewController.h

```
1. //
2. // DetailsViewController.h
3. // View_Switch
4. //
5. // Created by C00115609 on 28/01/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8.
9. #import <UIKit/UIKit.h>
10.
11. @class DetailsViewController;
12.
13. @protocol DetailsViewDelegate <NSObject>
14. @optional
15. - (void) maritalStatus: (DetailsViewController *)controller;
16.
17. @end
18.
19. @interface DetailsViewController : UIViewController <UIPickerViewDelegate, UIPickerViewDataSource>{
20.
21.     IBOutlet UIScrollView *scrollDetailsView;
22.     IBOutlet UILabel *cohabitingLabel;
23.     IBOutlet UISwitch *cohabitingSwitch;
24.     IBOutlet UIButton *cohabitingHelpButton;
25.     IBOutlet UIImageView *redOakManImageView;
26.     IBOutlet UIImageView *redOakManSmallerImageView;
27.     IBOutlet UIImageView *redOakLogoImageView;
```

```

28.     IBOutlet UISwitch *dependentChildrenSwitch;
29.
30.     IBOutlet UIButton *dateOfBirthPickerButton;
31.     IBOutlet UIToolbar *pickerToolBar;           // this toolbar appears only when the date picker is
open
32.     IBOutlet UIDatePicker *datePicker;
33.
34.     IBOutlet UIPickerView *pickerViewDetails;
35.     IBOutlet UIButton *maritalStatusPickerButton;
36.
37.     NSString *strMaritalStatus;
38.     NSDateFormatter *dateFormatter;
39.     UITextField *activeField;
40.     NSArray *pickerViewMaritalStatusArray;
41.
42.     id <DetailsViewDelegate> delegate;
43. }
44.
45. @property (assign) id <DetailsViewDelegate> delegate;
46.
47. @property (nonatomic, retain) IBOutlet UIPickerView *pickerViewDetails;
48. @property (nonatomic, retain) IBOutlet UIScrollView *scrollDetailsView;
49. @property (nonatomic, retain) IBOutlet UIDatePicker *datePicker;
50. @property (nonatomic, retain) IBOutlet UIToolbar *pickerToolBar;
51. @property (nonatomic, retain) IBOutlet UIButton *maritalStatusPickerButton;
52. @property (nonatomic, retain) IBOutlet UISwitch *dependentChildrenSwitch;
53. @property (nonatomic, retain) IBOutlet UISwitch *cohabitingSwitch;
54. @property (nonatomic, retain) IBOutlet UIButton *dateOfBirthPickerButton;
55.
56. @property (nonatomic, retain) NSDateFormatter *dateFormatter;
57. @property (nonatomic, assign) NSArray *pickerViewMaritalStatusArray;
58. @property (nonatomic, assign) NSString *strMaritalStatus;
59.
60. - (IBAction) maritalStatusHelpButtonPressed;
61. - (IBAction) dateOfBirthHelpButtonPressed;
62. - (IBAction) dependentChildrenHelpButtonPressed;

```

```
63. - (IBAction) cohabitingHelpButtonPressed;
64. - (IBAction) dateOfBirthPickerButtonPressed:(id) sender;
65.
66. - (IBAction) doneAction:(id) sender; // when the done button is clicked
67. - (IBAction) dateAction:(id) sender; // when the user has changed the date picker values (m/d/y)
68.
69. - (IBAction) dependentChildrenSwitchValueChanged:(id) sender;
70. - (IBAction) maritalStatusPickerButtonPressed:(id) sender;
71. - (IBAction) detailClosureButtonPressed:(id) sender;
72.
73. - (void) listenMaritalStatusDidChange;
74. - (void) popUpSlidePickerView;
75. - (void) popUpWithoutSlidePickerView;
76. - (void) popUpSlideDatePicker;
77. - (void) popUpWithoutSlideDatePicker;
78. - (void) resignDatePicker;
79. + (void) resignDatePicker;
80. - (void) resignPickerView;
81. + (void) resignPickerView;
82. - (void) presentCohabitingInfo;
83.
84. @end
```

2.4. DetailsViewController.m

```
1. //
2. // DetailsViewController.m
3. // View_Switch
4. //
5. // Created by C00115609 on 28/01/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8. /*****
9.                                     Class Description
10.    This class is responsible for controlling the "Details" view's functionalities.
11.    *****/
12.
13. #import "DetailsViewController.h"
14.
15. @implementation DetailsViewController
16.
17. @synthesize scrollDetailsView, dateFormatter, pickerToolBar, datePicker, pickerViewDetails;
18. @synthesize maritalStatusPickerButton, dependentChildrenSwitch, dateOfBirthPickerButton,
    cohabitingSwitch;
19. @synthesize pickerViewMaritalStatusArray, strMaritalStatus;
20. @synthesize delegate;
21.
22. // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
23. - (void)viewDidLoad {
24.     [super viewDidLoad];
25.
26.     scrollDetailsView.contentSize = CGSizeMake(320, 461);
27.
28.     self.dateFormatter = [[[NSDateFormatter alloc] init] autorelease];
29.     [self.dateFormatter setDateStyle:NSDateFormatterShortStyle];
30.     [self.dateFormatter setTimeStyle:NSDateFormatterNoStyle];
31.
32.     // Rent from items
33.     pickerViewMaritalStatusArray = [[NSArray alloc] initWithObjects:
```

```

34.                                     @"Single",
35.                                     @"Married",
36.                                     nil];
37.
38.     // Initialise the Date of Birth picker to "Single"
39.     maritalStatusPickerButton.titleLabel.text = [pickerViewMaritalStatusArray
objectAtIndex:[pickerViewDetails selectedRowInComponent:0]];
40.     strMaritalStatus = [[NSString alloc] initWithString: @"Single"];
41.
42.     // Change the Dependent Children Switch (Slider) text (YES/NO instead ON/OFF)
43.     [(UILabel *)[[[[[dependentChildrenSwitch subviews] lastObject] subviews] objectAtIndex:2]
subviews] objectAtIndex:0] setText:@"YES"];
44.     [(UILabel *)[[[[[dependentChildrenSwitch subviews] lastObject] subviews] objectAtIndex:2]
subviews] objectAtIndex:1] setText:@"NO"];
45.
46.     // Change the Cohabiting Switch (Slider) text (YES/NO instead ON/OFF)
47.     [(UILabel *)[[[[[cohabitingSwitch subviews] lastObject] subviews] objectAtIndex:2] subviews]
objectAtIndex:0] setText:@"YES"];
48.     [(UILabel *)[[[[[cohabitingSwitch subviews] lastObject] subviews] objectAtIndex:2] subviews]
objectAtIndex:1] setText:@"NO"];
49. }
50.
51. -(IBAction) maritalStatusHelpButtonPressed
52. {
53.     UIAlertView *alert = [[UIAlertView alloc]
54.                             initWithTitle:@"Tips: Marital Status"
55.                             message:@"The tax system can treat you differently depending on
your marital status. We'll check to see you get the best tax treatment."
56.                             delegate:nil
57.                             cancelButtonTitle:@"OK"
58.                             otherButtonTitles:nil];
59.     [alert show];
60.     [alert release];
61. }
62.
63. -(IBAction) dateOfBirthHelpButtonPressed

```

```
64.  {
65.      UIAlertView *alert = [[UIAlertView alloc]
66.                          initWithTitle:@"Tips: Date of Birth"
67.                          message:@"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."
68.                          delegate:nil
69.                          cancelButtonTitle:@"OK"
70.                          otherButtonTitles:nil];
71.      [alert show];
72.      [alert release];
73.  }
74.
75.  -(IBAction) dependentChildrenHelpButtonPressed
76.  {
77.      UIAlertView *alert = [[UIAlertView alloc]
78.                          initWithTitle:@"Tips: Dependent Children"
79.                          message:@"Dependent children are those still hanging around the
house - still at school or under 21 and in full time education."
80.                          delegate:nil
81.                          cancelButtonTitle:@"OK"
82.                          otherButtonTitles:nil];
83.      [alert show];
84.      [alert release];
85.  }
86.
87.  -(IBAction) cohabitingHelpButtonPressed
88.  {
89.      UIAlertView *alert = [[UIAlertView alloc]
90.                          initWithTitle:@"Tips: Are you co-habiting?"
91.                          message:@"If you are not married and have kids, we need to
check if you can be classed as a single parent by Revenue."
92.                          delegate:nil
93.                          cancelButtonTitle:@"OK"
94.                          otherButtonTitles:nil];
95.      [alert show];
96.      [alert release];
```



```

97.  }
98.
99.  #pragma mark -
100. #pragma mark UIPickerViewDataSource
101. - (NSString *)pickerView:(UIPickerView *)pickerView titleForRow:(NSInteger)row
    forComponent:(NSInteger)component
102. {
103.     return [pickerViewMaritalStatusArray objectAtIndex:row];
104. }
105.
106. - (NSInteger)pickerView:(UIPickerView *)pickerView numberOfRowsInComponent:(NSInteger)component
107. {
108.     return [pickerViewMaritalStatusArray count];
109. }
110.
111. - (NSInteger)numberOfComponentsInPickerView:(UIPickerView *)pickerView
112. {
113.     return 1;
114. }
115.
116. - (void)pickerView:(UIPickerView *)pickerView didSelectRow:(NSInteger)row
    inComponent:(NSInteger)component
117. {
118.     // Report the selection to the UI Button
119.     maritalStatusPickerButton.titleLabel.text = [pickerViewMaritalStatusArray
    objectAtIndex:[pickerViewDetails selectedRowInComponent:0]];
120.
121.     // Make invisible to cohabiting info
122.     [self presentCohabitingInfo];
123.
124.     if (![maritalStatusPickerButton titleLabel].text isEqualToString: strMaritalStatus) {
125.         [self listenMaritalStatusDidChange];
126.         strMaritalStatus = [[NSString alloc] initWithString:
    maritalStatusPickerButton.titleLabel.text];
127.     }
128. }

```

```
129.
130.
131. - (IBAction) dateOfBirthPickerButtonPressed:(id)sender
132. {
133.     // If keyboard is active then release it
134.     if(activeField != nil)
135.         [activeField resignFirstResponder];
136.
137.     // Remove the Picker View for marital status, if it is active
138.     if(self.pickerViewDetails.superview != nil){
139.         // Remove UIPickerView from superview
140.         [self.pickerViewDetails removeFromSuperview];
141.
142.         [self popUpWithoutSlideDatePicker];
143.
144.         // Report the selection to the UI Button
145.         maritalStatusPickerButton.titleLabel.text = [pickerViewMaritalStatusArray
146. objectAtIndex:[pickerViewDetails selectedRowInComponent:0]];
147.
148.         // Make invisible to cohabiting info
149.         [self presentCohabitingInfo];
150.     }
151.     // Check if date picker is already on screen
152.     else if (self.datePicker.superview == nil)
153.     {
154.         [self popUpSlideDatePicker];
155.     }
156. }
157.
158. // Show date picker with slide animation
159. -(void) popUpSlideDatePicker
160. {
161.     // Add the picker toolbar
162.     [self.view.window addSubview: self.pickerToolBar];
163.     //Add to picker view
```

```

164.     [self.view.window addSubview: self.datePicker];
165.
166.     // Size up the picker view to the screen and compute the start/end frame origin for the slide
    up animation
167.     //
168.     // Compute the start frame
169.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
170.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
171.     CGSize pickerSize = [self.datePicker sizeThatFits:CGSizeZero];
172.     CGRect startRectPickerToolBar = CGRectMake(0.0,
173.         screenRect.origin.y +
    screenRect.size.height,
174.         pickerToolBarSize.width,
    pickerToolBarSize.height);
175.     CGRect startRectPicker = CGRectMake(0.0,
176.         screenRect.origin.y +
    screenRect.size.height,
177.         pickerSize.width, pickerSize.height);
178.     self.pickerToolBar.frame = startRectPickerToolBar;
179.     self.datePicker.frame = startRectPicker;
180.
181.     // Compute the end frame
182.     CGRect pickerToolBarRect = CGRectMake(0.0,
183.         screenRect.origin.y +
    screenRect.size.height - (pickerToolBarSize.height + pickerSize.height),
184.         pickerToolBarSize.width,
185.         pickerToolBarSize.height);
186.
187.     // Compute the end frame
188.     CGRect pickerRect = CGRectMake(0.0,
189.         screenRect.origin.y + screenRect.size.height -
    pickerSize.height,
190.         pickerSize.width,
191.         pickerSize.height);
192.
193.     // Start the slide up animation

```

```

194.     [UIView beginAnimations:nil context:NULL];
195.     [UIView setAnimationDuration:0.3];
196.
197.     // Need to perform some post operations after the animation is complete
198.     [UIView setAnimationDelegate:self];
199.
200.     self.pickerToolBar.frame = pickerToolBarRect;
201.     self.datePicker.frame = pickerRect;
202.
203.     // Shrink the table vertical size to make room for the picker toolbar
204.     CGRect viewFrame = self.view.frame;
205.     viewFrame.size.height -= (self.pickerToolBar.frame.size.height);
206.     self.view.frame = viewFrame;
207.
208.     // Shrink the table vertical size to make room for the date picker
209.     viewFrame.size.height -= (self.datePicker.frame.size.height - 50);
210.     self.view.frame = viewFrame;
211.
212.     [UIView commitAnimations];
213. }
214.
215. // Show date picker without slide animation
216. -(void) popUpWithoutSlideDatePicker
217. {
218.     // Add the picker toolbar
219.     [self.view.window addSubview: self.pickerToolBar];
220.     //Add to picker view
221.     [self.view.window addSubview: self.datePicker];
222.
223.     // Size up the picker view to the screen and compute the start/end frame origin for the slide
    up animation
224.     //
225.     // Compute the start frame
226.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
227.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
228.     CGSize pickerSize = [self.datePicker sizeThatFits:CGSizeZero];

```

```

229.         CGRect startRectPickerToolbar = CGRectMake(0.0,
230.             screenRect.size.height,                               screenRect.origin.y +
231.             pickerToolBarSize.height);                             pickerToolBarSize.width,
232.         CGRect startRectPicker = CGRectMake(0.0,
233.             screenRect.size.height,                               screenRect.origin.y +
234.             pickerSize.width, pickerSize.height);
235.         self.pickerToolBar.frame = startRectPickerToolbar;
236.         self.datePicker.frame = startRectPicker;
237.
238.         // Compute the end frame
239.         CGRect pickerToolBarRect = CGRectMake(0.0,
240.             screenRect.size.height - (pickerToolBarSize.height + pickerSize.height), screenRect.origin.y +
241.             pickerToolBarSize.width,                               pickerToolBarSize.height);
242.
243.
244.         // Compute the end frame
245.         CGRect pickerRect = CGRectMake(0.0,
246.             pickerSize.height,                                     screenRect.origin.y + screenRect.size.height -
247.             pickerSize.width,                                     pickerSize.width,
248.             pickerSize.height);                                   pickerSize.height);
249.
250.         self.pickerToolBar.frame = pickerToolBarRect;
251.         self.datePicker.frame = pickerRect;
252.     }
253.
254. // The date picker has finished sliding downwards, so remove it
255. - (void)slideDownDatePickerDidStop
256. {
257.     [self.datePicker removeFromSuperview];
258. }
259.

```

```

260. // The picker view has finished sliding downwards, so remove it
261. - (void)slideDownPickerViewDetailsDidStop
262. {
263.     [self.pickerViewDetails removeFromSuperview];
264. }
265.
266. // Set the picker button value when the user changes date picker value
267. - (IBAction)dateAction:(id)sender
268. {
269.     dateOfBirthPickerButton.titleLabel.text = [self.dateFormatter
    stringFromDate:self.datePicker.date];
270. }
271.
272. + (void) resignPickerView{
273.     [self resignPickerView];
274. }
275.
276. + (void) resignDatePicker{
277.     [self resignDatePicker];
278. }
279.
280. - (IBAction) maritalStatusPickerButtonPressed:(id)sender{
281.     // Change the button label title to 'Select..'
282.     [maritalStatusPickerButton setTitle:@"Select..     " forState:UIControlStateNormal];
283.
284.     if(self.datePicker.superview != nil){
285.         // Remove datePicker fromSuperview
286.         [self.datePicker removeFromSuperview];
287.
288.         [self popUpWithoutSlidePickerView];
289.
290.         // Report the selection to the UI Button
291.         dateOfBirthPickerButton.titleLabel.text = [self.dateFormatter
    stringFromDate:self.datePicker.date];
292.     }
293.

```

```

294.         else {
295.             [self popUpSlidePickerView];
296.         }
297.     }
298.
299. // Show picker view with slide animation
300. - (void)popUpSlidePickerView{
301.
302.     // Check if date picker is on screen
303.     if (self.pickerViewDetails.superview == nil)
304.     {
305.         // Add the picker toolbar
306.         [self.view.window addSubview: self.pickerToolBar];
307.         //Add to picker view
308.         [self.view.window addSubview: self.pickerViewDetails];
309.
310.         // Size up the picker view to the screen and compute the start/end frame origin for the
            slide up animation
311.         //
312.         // Compute the start frame
313.         CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
314.         CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
315.         CGSize pickerSize = [self.pickerViewDetails sizeThatFits:CGSizeZero];
316.         CGRect startRectPickerToolBar = CGRectMake(0.0,
317.             screenRect.origin.y +
            screenRect.size.height,
318.             pickerToolBarSize.width,
            pickerToolBarSize.height);
319.         CGRect startRectPicker = CGRectMake(0.0,
320.             screenRect.origin.y +
            screenRect.size.height,
321.             pickerSize.width,
            pickerSize.height);
322.         self.pickerToolBar.frame = startRectPickerToolBar;
323.         self.pickerViewDetails.frame = startRectPicker;
324.

```

```

325.         // Compute the end frame
326.         CGRect pickerToolBarRect = CGRectMake(0.0,
327.         screenRect.origin.y +
screenRect.size.height - (pickerToolBarSize.height + pickerSize.height),
328.         pickerToolBarSize.width,
329.         pickerToolBarSize.height);
330.
331.         // Compute the end frame
332.         CGRect pickerRect = CGRectMake(0.0,
333.         screenRect.origin.y + screenRect.size.height
- pickerSize.height,
334.         pickerSize.width,
335.         pickerSize.height);
336.
337.         // Start the slide up animation
338.         [UIView beginAnimations:nil context:NULL];
339.         [UIView setAnimationDuration:0.3];
340.
341.         // Need to perform some post operations after the animation is complete
342.         [UIView setAnimationDelegate:self];
343.
344.         self.pickerToolBar.frame = pickerToolBarRect;
345.         self.pickerViewDetails.frame = pickerRect;
346.
347.         // Shrink the table vertical size to make room for the picker toolbar
348.         CGRect viewFrame = self.view.frame;
349.         viewFrame.size.height -= (self.pickerToolBar.frame.size.height);
350.
351.         // Shrink the table vertical size to make room for the date picker
352.         viewFrame.size.height -= (self.pickerViewDetails.frame.size.height - 50);
353.         self.view.frame = viewFrame;
354.
355.         [UIView commitAnimations];
356.
357.         // Scroll the active picker button into view.
358.         CGRect maritalStatusPickerButtonRect = [maritalStatusPickerButton frame];

```



```

359.         [scrollDetailsView scrollRectToVisible:maritalStatusPickerButtonRect animated:YES];
360.     }
361. }
362.
363. // Show picker view without slide animation
364. -(void) popUpWithoutSlidePickerView
365. {
366.     // Add the picker toolbar
367.     [self.view.window addSubview: self.pickerToolBar];
368.     //Add to picker view
369.     [self.view.window addSubview: self.pickerViewDetails];
370.
371.     // Size up the picker view to the screen and compute the start/end frame origin for the slide
    up animation
372.     //
373.     // Compute the start frame
374.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
375.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
376.     CGSize pickerSize = [self.pickerViewDetails sizeThatFits:CGSizeZero];
377.     CGRect startRectPickerToolBar = CGRectMake(0.0,
378.                                                screenRect.origin.y +
    screenRect.size.height,
379.                                                pickerToolBarSize.width,
    pickerToolBarSize.height);
380.     CGRect startRectPicker = CGRectMake(0.0,
381.                                        screenRect.origin.y +
    screenRect.size.height,
382.                                        pickerSize.width, pickerSize.height);
383.     self.pickerToolBar.frame = startRectPickerToolBar;
384.     self.pickerViewDetails.frame = startRectPicker;
385.
386.     // Compute the end frame
387.     CGRect pickerToolBarRect = CGRectMake(0.0,
388.                                           screenRect.origin.y +
    screenRect.size.height - (pickerToolBarSize.height + pickerSize.height),
389.                                           pickerToolBarSize.width,

```

```

390.                                     pickerToolBarSize.height);
391.
392.     // Compute the end frame
393.     CGRect pickerRect = CGRectMake(0.0,
394.     pickerSize.height,                                     screenRect.origin.y + screenRect.size.height -
395.                                     pickerSize.width,
396.                                     pickerSize.height);
397.
398.     self.pickerToolBar.frame = pickerToolBarRect;
399.     self.pickerViewDetails.frame = pickerRect;
400. }
401.
402.
403. - (void) resignPickerView{
404.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
405.     CGRect endFrameToolBar = self.pickerViewDetails.frame;
406.     CGRect endFramePicker = self.pickerViewDetails.frame;
407.     endFrameToolBar.origin.y = screenRect.origin.y + screenRect.size.height;
408.     endFramePicker.origin.y = screenRect.origin.y + screenRect.size.height;
409.
410.     // Start the slide down animation
411.     [UIView beginAnimations:nil context:NULL];
412.     [UIView setAnimationDuration:0.3];
413.
414.     // Need to perform some post operations after the animation is complete
415.     [UIView setAnimationDelegate:self];
416.     [UIView setAnimationDidStopSelector:@selector(slideDownPickerViewDetailsDidStop)];
417.
418.     self.pickerToolBar.frame = endFrameToolBar;
419.     self.pickerViewDetails.frame = endFramePicker;
420.     [UIView commitAnimations];
421.
422.     // Grow the table back again in vertical size to make room for the date picker
423.     CGRect viewFrame = self.view.frame;
424.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];

```

```

425.         CGSize pickerSize = [self.pickerViewDetails sizeThatFits:CGSizeZero];
426.         viewFrame.size.height += (self.pickerToolBar.frame.size.height - (pickerSize.height -
    pickerToolBarSize.height));
427.         viewFrame.size.height += (self.pickerViewDetails.frame.size.height - 50);
428.         self.view.frame = viewFrame;
429.
430.         // Report the selection to the UI Button
431.         maritalStatusPickerButton.titleLabel.text = [pickerViewMaritalStatusArray
    objectAtIndex:[pickerViewDetails selectedRowInComponent:0]];
432.     }
433.
434. - (void) resignDatePicker{
435.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
436.     CGRect endFrameToolBar = self.datePicker.frame;
437.     CGRect endFramePicker = self.datePicker.frame;
438.     endFrameToolBar.origin.y = screenRect.origin.y + screenRect.size.height;
439.     endFramePicker.origin.y = screenRect.origin.y + screenRect.size.height;
440.
441.     // start the slide down animation
442.     [UIView beginAnimations:nil context:NULL];
443.     [UIView setAnimationDuration:0.3];
444.
445.     // we need to perform some post operations after the animation is complete
446.     [UIView setAnimationDelegate:self];
447.     [UIView setAnimationDidStopSelector:@selector(slideDownDatePickerDidStop)];
448.
449.     self.pickerToolBar.frame = endFrameToolBar;
450.     self.datePicker.frame = endFramePicker;
451.     [UIView commitAnimations];
452.
453.     // grow the table back again in vertical size to make room for the date picker
454.     CGRect viewFrame = self.view.frame;
455.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
456.     CGSize pickerSize = [self.datePicker sizeThatFits:CGSizeZero];
457.     viewFrame.size.height += (self.pickerToolBar.frame.size.height - (pickerSize.height -
    pickerToolBarSize.height));

```

```

458.         viewFrame.size.height += (self.datePicker.frame.size.height - 50);
459.         self.view.frame = viewFrame;
460.
461.         // Report the selection to the UI Button
462.         dateOfBirthPickerButton.titleLabel.text = [self.dateFormatter
stringFromDate:self.datePicker.date];
463.     }
464.
465. - (IBAction)doneAction:(id)sender{
466.     // Check if date picker is already on screen
467.     if (self.datePicker.superview != nil)
468.         [self resignDatePicker];
469.     else
470.         [self resignPickerView];
471. }
472.
473. - (IBAction) dependentChildrenSwitchValueChanged:(id)sender{
474.
475.     // If marital status picker view is active then release it
476.     if (self.pickerViewDetails.superview != nil) {
477.         [self resignPickerView];
478.     }
479.     // If picker view is active then release it
480.     else if (self.datePicker.superview != nil) {
481.         [self resignDatePicker];
482.     }
483.
484.     // Make visible or invisible to cohabiting info
485.     [self presentCohabitingInfo];
486. }
487.
488. - (void) presentCohabitingInfo {
489.
490.     NSString *trimmedString = [[[maritalStatusPickerButton titleLabel] text]
stringByTrimmingCharactersInSet:[NSCharacterSet
whitespaceAndNewlineCharacterSet]];

```

```

492.     //If the user is single and has dependent children then present the cohabiting info
493.     if ([trimmedString isEqualToString:@"Single"] && dependentChildrenSwitch.on == YES) {
494.         cohabitingLabel.hidden = NO;
495.         cohabitingSwitch.hidden = NO;
496.         cohabitingHelpButton.hidden = NO;
497.         redOakManSmallerImageView.hidden = NO;
498.         redOakManImageView.hidden = YES;
499.     }
500.     //Not married or has dependent children
501.     else if(cohabitingLabel.hidden == NO){
502.         cohabitingLabel.hidden = YES;
503.         cohabitingSwitch.hidden = YES;
504.         cohabitingHelpButton.hidden = YES;
505.         redOakManSmallerImageView.hidden = YES;
506.         redOakManImageView.hidden = NO;
507.     }
508. }
509.
510. - (void) listenMaritalStatusDidChange {
511.     if ([self.delegate respondsToSelector:@selector(maritalStatus:)]) {
512.         [delegate maritalStatus: self];
513.     }
514. }
515.
516. // Shutdown the app and go to Red Oak Company's website
517. - (IBAction) detailClosureButtonPressed:(id)sender {
518.     [[UIApplication sharedApplication] openURL:[NSURL
519.         URLWithString:@"http://www.redoaktaxrefunds.ie"]];
520. }
521. /*
522. // The designated initializer. Override if you create the controller programmatically and want to
523. // perform customization that is not appropriate for viewDidLoad.
524. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
525.     if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
526.         // Custom initialization

```

```
526. }
527. return self;
528. }
529. */
530.
531. /*
532. // Implement loadView to create a view hierarchy programmatically, without using a nib.
533. - (void)loadView {
534. }
535. */
536.
537. /*
538. // Override to allow orientations other than the default portrait orientation.
539. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
540. // Return YES for supported orientations
541. return (interfaceOrientation == UIInterfaceOrientationPortrait);
542. }
543. */
544.
545. - (void)didReceiveMemoryWarning {
546.     // Releases the view if it doesn't have a superview.
547.     [super didReceiveMemoryWarning];
548.
549.     // Release any cached data, images, etc that aren't in use.
550. }
551.
552. - (void)viewDidUnload {
553.     self.pickerToolBar = nil;
554.     self.dateFormatter = nil;
555.     // Release any retained subviews of the main view.
556.     // e.g. self.myOutlet = nil;
557. }
558.
559. - (void)dealloc {
560.     [scrollDetailsView release];
561.     [datePicker release];
```

```
562.     [dateFormatter release];
563.     [pickerViewDetails release];
564.     [pickerViewMaritalStatusArray release];
565.     [strMaritalStatus release];
566.     [pickerToolBar release];
567.     [maritalStatusPickerButton release];
568.     [dependentChildrenSwitch release];
569.     [dateOfBirthPickerButton release];
570.     [delegate release];
571.     [super dealloc];
572. }
573.
574. @end
```

2.5. MyTaxViewController.h

```
1. //
2. // MyTaxViewController.h
3. // View_Switch
4. //
5. // Created by C00115609 on 28/01/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8.
9. #import <UIKit/UIKit.h>
10.
11. @interface MyTaxViewController : UIViewController {
12.
13.     IBOutlet UIScrollView *scrollMyTaxView;
14.     IBOutlet UITextField *annualSalaryTextField;
15.     IBOutlet UITextField *taxOnPayTextField;
16.     IBOutlet UISwitch *tradeUnionSwitch;
17.     UITextField *activeField;
18. }
19.
```

```
20. @property (nonatomic, retain) IBOutlet UIScrollView *scrollMyTaxView;
21. @property (nonatomic, retain) IBOutlet UITextField *annualSalaryTextField;
22. @property (nonatomic, retain) IBOutlet UITextField *taxOnPayTextField;
23. @property (nonatomic, retain) IBOutlet UISwitch *tradeUnionSwitch;
24.
25. @property (nonatomic, assign) IBOutlet UITextField *activeField;
26.
27. - (IBAction) anualSalaryBeforeTaxHelpButtonPressed;
28. - (IBAction) TaxOnPayHelpButtonPressed;
29. - (IBAction) MemberTradeUnionHelpButtonPressed;
30. - (IBAction) textFieldDoneEditing:(id) sender;
31. - (IBAction) tradeUnionMemberSwitchValueChanged;
32. - (IBAction) detailClosureButtonPressed:(id) sender;
33.
34. - (void) textFieldDidBeginEditing:(UITextField *)textField;
35. - (void) textFieldDidEndEditing:(UITextField *)textField;
36. - (void) limitTextField:(NSNotification *)notif;
37.
38. - (NSString*) commaSeparateThousands: (int) number;
39.
40. @end
```


2.6. MyTaxViewController.m

```
1. //
2. // MyTaxViewController.m
3. // View_Switch
4. //
5. // Created by C00115609 on 28/01/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8. /*****
9.                                     Class Description
10.    This class is responsible for controlling the "My Tax" view's functionalities.
11.    *****/
12.
13. #import "MyTaxViewController.h"
14.
15. @implementation MyTaxViewController
16.
17. @synthesize annualSalaryTextField, taxOnPayTextField, tradeUnionSwitch, scrollMyTaxView, activeField;
18.
19. // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
20. - (void)viewDidLoad {
21.     [super viewDidLoad];
22.
23.     scrollMyTaxView.contentSize = CGSizeMake(320, 461);
24.
25.     // Change the Trade Union Switch (Slider) text (YES/NO instead ON/OFF)
26.     [(UILabel *)[[[tradeUnionSwitch subviews] lastObject] subviews] objectAtIndex:2] subviews]
        objectAtIndex:0] setText:@"YES"];
27.     [(UILabel *)[[[tradeUnionSwitch subviews] lastObject] subviews] objectAtIndex:2] subviews]
        objectAtIndex:1] setText:@"NO"];
28. }
29.
30. -(IBAction) anualSalaryBeforeTaxHelpButtonPressed
31. {
32.     UIAlertView *alert = [[UIAlertView alloc]
```

```

33.             initWithTitle:@"Tips: Annual Salary before Tax"
34.             message:@"What's your annual salary before tax and after
pension deductions. This is the gross pay from your P60 (or P45)"
35.             delegate:nil
36.             cancelButtonTitle:@"OK"
37.             otherButtonTitles:nil];
38.     [alert show];
39.     [alert release];
40. }
41.
42. - (IBAction) TaxOnPayHelpButtonPressed
43. {
44.     UIAlertView *alert = [[UIAlertView alloc]
45.         initWithTitle:@"Tips: Tax on Salary"
46.         message:@"How much tax did you pay on that salary? Don't
include other deductions like PRSI or income levy here, just income tax."
47.         delegate:nil
48.         cancelButtonTitle:@"OK"
49.         otherButtonTitles:nil];
50.     [alert show];
51.     [alert release];
52. }
53.
54. - (IBAction) MemberTradeUnionHelpButtonPressed
55. {
56.     UIAlertView *alert = [[UIAlertView alloc]
57.         initWithTitle:@"Tips: Member of a Trade Union?"
58.         message:@"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."
59.         delegate:nil
60.         cancelButtonTitle:@"OK"
61.         otherButtonTitles:nil];
62.     [alert show];
63.     [alert release];
64. }
65.

```

```

66. -(IBAction) tradeUnionMemberSwitchValueChanged
67. {
68.     // If keyboard is active then release it
69.     if(activeField != nil)
70.         [activeField resignFirstResponder];
71. }
72.
73. /* Resign the keyboard */
74. -(IBAction) textFieldDoneEditing:(id)sender{
75.     [sender resignFirstResponder];
76. }
77.
78. - (void)textFieldDidBeginEditing:(UITextField *)textField
79. {
80.     activeField = textField;
81. }
82.
83. - (void)textFieldDidEndEditing:(UITextField *)textField
84. {
85.     // When editing is finished, separate number into fractional parts
86.     [textField setText:[self commaSeparateThousands: [[textField text] intValue]]];
87.     activeField = nil;
88. }
89.
90. // Thousand Separator
91. -(NSString*) commaSeparateThousands: (int) number
92. {
93.     NSString *result = @"";
94.     while( number > 0 )
95.     {
96.         int removedDigits = number % 1000;
97.         if( result != nil && [result length] > 0 )
98.         {
99.             if( number > 999 )
100.                result = [NSString stringWithFormat:@"%i.%03i,%@", removedDigits, result];
101.             else

```

```

102.             result = [NSString stringWithFormat:@"%i,%@", removedDigits, result];
103.         }
104.         else
105.         {
106.             if( number > 999 )
107.                 result = [NSString stringWithFormat:@"%i.%03i", removedDigits];
108.             else
109.                 result = [NSString stringWithFormat:@"%i", removedDigits];
110.         }
111.         number = number/1000;
112.     }
113.
114.     return result;
115. }
116.
117.
118. - (void)viewWillAppear:(BOOL)animated{
119.     [super viewWillAppear:animated];
120.     // Registering for text field events
121.     [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(limitTextField:)
122.     name:@"UITextFieldTextDidChangeNotification" object:activeField];
123. }
124.
125. - (void)viewWillDisappear:(BOOL)animated{
126.     // Unregistering for text field events
127.     [[NSNotificationCenter defaultCenter] removeObserver:self
128.     name:@"UITextFieldTextDidChangeNotification" object:activeField];
129. }
130. // Limit the text field input
131. - (void)limitTextField:(NSNotification *)notif {
132.     int limit = 9;
133.     if ([[activeField text] length] > limit) {
134.         [activeField setText:[activeField text] substringToIndex:limit]];
135.     }

```

```
136. }
137.
138. // Shutdown the app and go to Red Oak Company's website
139. - (IBAction) detailClosureButtonPressed:(id)sender {
140.     [[UIApplication sharedApplication] openURL:[NSURL
141.         URLWithString:@"http://www.redoaktaxrefunds.ie"]];
142. }
143. /*
144. // The designated initializer. Override if you create the controller programmatically and want to
145. // perform customization that is not appropriate for viewDidLoad.
146. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
147.     if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
148.         // Custom initialization
149.     }
150.     return self;
151. }
152. */
153. /*
154. // Implement loadView to create a view hierarchy programmatically, without using a nib.
155. - (void)loadView {
156. }
157. */
158.
159. /*
160. // Override to allow orientations other than the default portrait orientation.
161. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
162.     // Return YES for supported orientations
163.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
164. }
165. */
166.
167. - (void)didReceiveMemoryWarning {
168.     // Releases the view if it doesn't have a superview.
169.     [super didReceiveMemoryWarning];
```

```
170.
171.     // Release any cached data, images, etc that aren't in use.
172. }
173.
174. - (void)viewDidUnload {
175.     // Release any retained subviews of the main view.
176.     // e.g. self.myOutlet = nil;
177. }
178.
179.
180. - (void)dealloc {
181.     [scrollMyTaxView release];
182.     [annualSalaryTextField release];
183.     [taxOnPayTextField release];
184.     [tradeUnionSwitch release];
185.     [super dealloc];
186. }
187.
188. @end
```

2.7. ExpensesViewController.h

```
1. //
2. // ExpensesViewController.h
3. // View_Switch
4. //
5. // Created by C00115609 on 28/01/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8.
9. #import <UIKit/UIKit.h>
10.
11. @interface ExpensesViewController : UIViewController <UITextFieldDelegate, UIPickerViewDelegate,
    UIPickerViewDataSource>
12. {
13.     IBOutlet UIScrollView *scrollExpensesView;
14.     IBOutlet UIPickerView *pickerViewRentOrMortgage;
15.     IBOutlet UIPickerView *pickerViewRentFrom;
16.     IBOutlet UIToolbar *pickerToolBar;           // this toolbar appears only when the date picker is
    open
17.     IBOutlet UIButton *rentOrMorgagePickerButton;
18.
19.     IBOutlet UILabel *rentFromLabel;
20.     IBOutlet UIButton *rentFromPickerButton;
21.     IBOutlet UIButton *rentFromHelpButton;
22.
23.     IBOutlet UILabel *annualRentLabel;
24.     IBOutlet UILabel *annualRentEuroLabel;
25.     IBOutlet UITextField *annualRentTextField;
26.     IBOutlet UIButton *annualRentHelpButton;
27.
28.     IBOutlet UILabel *interestReliefLabel;
29.     IBOutlet UISwitch *interestReliefSwitch;
30.     IBOutlet UIButton *interestReliefHelpButton;
31.
32.     IBOutlet UIImageView *redOakManImageView;
```

```

33.     IBOutlet UIImageView *blankImageView;
34.     IBOutlet UIImageView *pickerDownArrowImageView;
35.
36.     IBOutlet UITextField *allowableMedicalTextField;
37.     IBOutlet UITextField *allowableDentalTextField;
38.     IBOutlet UITextField *binChargesTextField;
39.
40.     NSArray *pickerViewRentOrMortgageArray;
41.     NSArray *pickerViewRentFromArray;
42.     UITextField *activeField;
43.     BOOL keyboardVisible;
44.     BOOL rentOrMortgageFlag;
45. }
46.
47. @property (nonatomic, retain) IBOutlet UIScrollView *scrollExpensesView;
48. @property (nonatomic, retain) IBOutlet UIPickerView *pickerViewRentOrMortgage;
49. @property (nonatomic, retain) IBOutlet UIPickerView *pickerViewRentFrom;
50. @property (nonatomic, retain) IBOutlet UIToolbar *pickerToolBar;
51. @property (nonatomic, retain) IBOutlet UIButton *rentOrMortgagePickerButton;
52. @property (nonatomic, retain) IBOutlet UIButton *rentFromPickerButton;
53.
54. @property (nonatomic, retain) IBOutlet UITextField *allowableMedicalTextField;
55. @property (nonatomic, retain) IBOutlet UITextField *allowableDentalTextField;
56. @property (nonatomic, retain) IBOutlet UITextField *binChargesTextField;
57. @property (nonatomic, retain) IBOutlet UITextField *annualRentTextField;
58. @property (nonatomic, retain) IBOutlet UISwitch *interestReliefSwitch;
59.
60. @property (nonatomic, retain) IBOutlet UILabel *rentFromLabel;
61. @property (nonatomic, retain) IBOutlet UIImageView *blankImageView;
62. @property (nonatomic, retain) IBOutlet UIImageView *pickerDownArrowImageView;
63. @property (nonatomic, retain) IBOutlet UIButton *rentFromHelpButton;
64.
65. @property (nonatomic, retain) IBOutlet UILabel *annualRentLabel;
66. @property (nonatomic, retain) IBOutlet UILabel *annualRentEuroLabel;
67. @property (nonatomic, retain) IBOutlet UIButton *annualRentHelpButton;
68.

```



```
69. @property (nonatomic, retain) IBOutlet UILabel *interestReliefLabel;
70. @property (nonatomic, retain) IBOutlet UIButton *interestReliefHelpButton;
71.
72. @property (nonatomic, retain) IBOutlet UIImageView *redOakManImageView;
73.
74. @property (nonatomic, assign) BOOL keyboardVisible;
75. @property (nonatomic, assign) NSArray *pickerViewRentOrMortgageArray;
76.
77. - (IBAction) allowableMedicalExpensesHelpButtonPressed;
78. - (IBAction) allowableDentalExpensesHelpButtonPressed;
79. - (IBAction) binChargesPrevYearHelpButtonPressed;
80. - (IBAction) rentMorgageHelpButtonPressed;
81. - (IBAction) interestReliefHelpButtonPressed;
82. - (IBAction) interestReliefSwitchValueChanged;
83. - (IBAction) rentFromHelpButtonPressed;
84. - (IBAction) annualRentHelpButtonPressed;
85. - (IBAction) textFieldDoneEditing:(id) sender;
86. - (IBAction) rentOrMortgagePickerButtonPressed:(id) sender;
87. - (IBAction) rentFromPickerButtonPressed:(id) sender;
88.
89. - (IBAction) doneAction:(id) sender; // when the done button is clicked
90.
91. - (void) keyboardDidShow:(NSNotification *)notif;
92. - (void) keyboardWillHide:(NSNotification *)notif;
93. - (void) textFieldDidBeginEditing:(UITextField *)textField;
94. - (void) textFieldDidEndEditing:(UITextField *)textField;
95.
96. - (void) popUpSlidePickerViewRentOrMortgage;
97. - (void) popUpWithoutSlidePickerViewRentOrMortgage;
98. - (void) popUpSlidePickerViewRentOrMortgage;
99. - (void) popUpSlidePickerViewRentFrom;
100. - (void) popUpWithoutSlidePickerViewRentFrom;
101. - (void) resignPickerViewRentOrMortgage;
102. - (void) resignPickerViewRentFrom;
103. + (void) resignPickerView;
104. - (void) resignKeyboard;
```

```

105. + (void) resignKeyboard;
106.
107. - (void) addSubRentOrMortgageInfo;
108. - (void) limitTextField:(NSNotification *)notif;
109.
110. - (NSString*) commaSeparateThousands: (int) number;
111.
112. @end

```

2.8. ExpensesViewController.m

```

1. //
2. // ExpensesViewController.m
3. // View_Switch
4. //
5. // Created by C00115609 on 28/01/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8. /*****
9.             Class Description
10.    This class is responsible for controlling the "Expenses" view's functionalities.
11.    *****/
12.
13. #import "ExpensesViewController.h"
14.
15. @implementation ExpensesViewController
16. @synthesize pickerViewRentOrMortgage, pickerViewRentFrom, scrollExpensesView, pickerToolBar;
17. @synthesize rentOrMorgagePickerButton, rentFromPickerButton, allowableMedicalTextField,
    allowableDentalTextField,
18. binChargesTextField, keyboardVisible, annualRentTextField, interestReliefSwitch;
19. @synthesize rentFromLabel, blankImageView, pickerDownArrowImageView, rentFromHelpButton,
20. annualRentLabel, annualRentEuroLabel, annualRentHelpButton, interestReliefLabel,
    interestReliefHelpButton,
21. redOakManImageView, pickerViewRentOrMortgageArray;
22.

```

```

23. // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
24. - (void)viewDidLoad {
25.     [super viewDidLoad];
26.
27.     scrollExpensesView.contentSize = CGSizeMake(320, 461);
28.
29.     rentOrMortgageFlag = YES;
30.     // this list appears in the UIPickerView to pick
31.     // Rent or Mortgage items
32.     pickerViewRentOrMortgageArray = [[NSArray alloc] initWithObjects:
33.                                     @"Neither",
34.                                     @"Rent",
35.                                     @"Mortgage",
36.                                     nil];
37.
38.     // Rent from items
39.     pickerViewRentFromArray = [[NSArray alloc] initWithObjects:
40.                                 @"Private landloard",
41.                                 @"Housing authority",
42.                                 @"Parent",
43.                                 nil];
44.
45.     // Initialise the Rent Or Mortgage picker to "Neither"
46.     rentOrMorgagePickerButton.titleLabel.text = [pickerViewRentOrMortgageArray
47. objectAtIndex:[pickerViewRentOrMortgage selectedRowInComponent:0]];
48.     // Change the Interest Relief Switch (Slider) text (YES/NO instead ON/OFF)
49.     [(UILabel *)[[[interestReliefSwitch subviews] lastObject] subviews] objectAtIndex:2]
50.     subviews] objectAtIndex:0] setText:@"YES"];
51.     [(UILabel *)[[[interestReliefSwitch subviews] lastObject] subviews] objectAtIndex:2]
52.     subviews] objectAtIndex:1] setText:@"NO"];
53. }
54. - (IBAction) allowableMedicalExpensesHelpButtonPressed
55. {
    UIAlertView *alert = [[UIAlertView alloc]

```

```

56.             initWithTitle:@"Tips: Allowable Medical Expenses"
57.             message:@"What's allowable? Generally doctor expenses and
    drugs and treatments a doctor prescribes."
58.             delegate:nil
59.             cancelButtonTitle:@"OK"
60.             otherButtonTitles:nil];
61.     [alert show];
62.     [alert release];
63. }
64.
65. - (IBAction) allowableDentalExpensesHelpButtonPressed
66. {
67.     UIAlertView *alert = [[UIAlertView alloc]
68.         initWithTitle:@"Tips: Allowable Dental Expenses"
69.         message:@"Ordinary dentist visits are not allowed (you should
    get them through your PRSI). Claim dental surgery here."
70.         delegate:nil
71.         cancelButtonTitle:@"OK"
72.         otherButtonTitles:nil];
73.     [alert show];
74.     [alert release];
75. }
76.
77. - (IBAction) binChargesPrevYearHelpButtonPressed
78. {
79.     UIAlertView *alert = [[UIAlertView alloc]
80.         initWithTitle:@"Tips: Bin charges for prev year"
81.         message:@"Paying for bins? Keep the receipts or records and
    claim a tax refund for this."
82.         delegate:nil
83.         cancelButtonTitle:@"OK"
84.         otherButtonTitles:nil];
85.     [alert show];
86.     [alert release];
87. }
88.

```

```
89. - (IBAction) rentMorgageHelpButtonPressed
90. {
91.     UIAlertView *alert = [[UIAlertView alloc]
92.         initWithTitle:@"Tips: Rent or Morgage"
93.         message:@"If you are renting or have a mortgage you can be due
tax back, select which applies to you."
94.         delegate:nil
95.         cancelButtonTitle:@"OK"
96.         otherButtonTitles:nil];
97.     [alert show];
98.     [alert release];
99. }
100.
101. - (IBAction) interestReliefHelpButtonPressed
102. {
103.     UIAlertView *alert = [[UIAlertView alloc]
104.         initWithTitle:@"Tips: Morgage interest relief?"
105.         message:@"Many people who were receiving mortgage interest
relief lost it in the last year - Are you due mortgage interest relief?"
106.         delegate:nil
107.         cancelButtonTitle:@"OK"
108.         otherButtonTitles:nil];
109.     [alert show];
110.     [alert release];
111. }
112.
113. - (IBAction) rentFromHelpButtonPressed
114. {
115.     UIAlertView *alert = [[UIAlertView alloc]
116.         initWithTitle:@"Tips: Who do you rent from?"
117.         message:@"If you are renting from a private landlord you are
due rent relief - but not if you are renting from a parent / housing authority"
118.         delegate:nil
119.         cancelButtonTitle:@"OK"
120.         otherButtonTitles:nil];
121.     [alert show];
```

```

122.     [alert release];
123. }
124.
125. - (IBAction) annualRentHelpButtonPressed
126. {
127.     UIAlertView *alert = [[UIAlertView alloc]
128.         initWithTitle:@"Tips: What is your annual rent?"
129.         message:@"How much rent are you paying - this can impact the
        amount of rent relief you can claim."
130.         delegate:nil
131.         cancelButtonTitle:@"OK"
132.         otherButtonTitles:nil];
133.     [alert show];
134.     [alert release];
135. }
136.
137. /* Resign the keyboard */
138. - (IBAction) textFieldDoneEditing:(id)sender{
139.     [sender resignFirstResponder];
140. }
141.
142. #pragma mark -
143. #pragma mark UIPickerViewDataSource
144. - (NSString *)pickerView:(UIPickerView *)pickerView titleForRow:(NSInteger)row
    forComponent:(NSInteger)component
145. {
146.     if (rentOrMortgageFlag) {
147.         return [pickerViewRentOrMortgageArray objectAtIndex:row];
148.     }
149.     else {
150.         return [pickerViewRentFromArray objectAtIndex:row];
151.     }
152.
153.
154. }
155.

```

```
156. - (NSInteger)pickerView:(UIPickerView *)pickerView numberOfRowsInComponent:(NSInteger) component
157. {
158.     if (rentOrMortgageFlag) {
159.         return [pickerViewRentOrMortgageArray count];
160.     }
161.     else {
162.         return [pickerViewRentFromArray count];
163.     }
164. }
165.
166. - (NSInteger)numberOfComponentsInPickerView:(UIPickerView *)pickerView
167. {
168.     return 1;
169. }
170.
171. - (void)pickerView:(UIPickerView *)pickerView didSelectRow:(NSInteger) row
    inComponent:(NSInteger) component
172. {
173.     if (rentOrMortgageFlag) {
174.         // Report the selection to the UI Button
175.         rentOrMortgagePickerButton.titleLabel.text = [pickerViewRentOrMortgageArray
    objectAtIndex:[pickerViewRentOrMortgage selectedRowInComponent:0]];
176.
177.         if (rentFromPickerButton.hidden || ![[rentOrMortgagePickerButton titleLabel] text]
    isEqualToString:@"Rent"]) {
178.             [self addSubRentOrMortgageInfo];
179.         }
180.     }
181.     else {
182.         rentFromPickerButton.titleLabel.text = [pickerViewRentFromArray
    objectAtIndex:[pickerViewRentFrom selectedRowInComponent:0]];
183.     }
184.
185. }
186.
187. // Show picker view rent or mortgage with slide animation
```

```

188. - (void)popUpSlidePickerViewRentOrMortgage
189. {
190.     // If keyboard is active then release it
191.     if(activeField != nil)
192.         [activeField resignFirstResponder];
193.
194.     // Check if date picker is already on screen
195.     if (self.pickerViewRentOrMortgage.superview == nil)
196.     {
197.         // Add the picker toolbar
198.         [self.view.window addSubview: self.pickerToolBar];
199.         //Add to picker view
200.         [self.view.window addSubview: self.pickerViewRentOrMortgage];
201.
202.         // Size up the picker view to the screen and compute the start/end frame origin for the
         slide up animation
203.         //
204.         // Compute the start frame
205.         CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
206.         CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
207.         CGSize pickerSize = [self.pickerViewRentOrMortgage sizeThatFits:CGSizeZero];
208.         CGRect startRectPickerToolBar = CGRectMake(0.0,
209.                                                     screenRect.origin.y +
         screenRect.size.height,
210.                                                     pickerToolBarSize.width,
         pickerToolBarSize.height);
211.         CGRect startRectPicker = CGRectMake(0.0,
212.                                             screenRect.origin.y +
         screenRect.size.height,
213.                                             pickerSize.width,
         pickerSize.height);
214.         self.pickerToolBar.frame = startRectPickerToolBar;
215.         self.pickerViewRentOrMortgage.frame = startRectPicker;
216.
217.         // Compute the end frame
218.         CGRect pickerToolBarRect = CGRectMake(0.0,

```



```

219.                                     screenRect.origin.y +
    screenRect.size.height - (pickerToolBarSize.height + pickerSize.height),
220.                                     pickerToolBarSize.width,
221.                                     pickerToolBarSize.height);
222.
223.         // Compute the end frame
224.         CGRect pickerRect = CGRectMake(0.0,
225.                                     screenRect.origin.y + screenRect.size.height
    - pickerSize.height,
226.                                     pickerSize.width,
227.                                     pickerSize.height);
228.
229.         // Start the slide up animation
230.         [UIView beginAnimations:nil context:NULL];
231.         [UIView setAnimationDuration:0.3];
232.
233.         // Need to perform some post operations after the animation is complete
234.         [UIView setAnimationDelegate:self];
235.
236.         self.pickerToolBar.frame = pickerToolBarRect;
237.         self.pickerViewRentOrMortgage.frame = pickerRect;
238.
239.         // Shrink the table vertical size to make room for the picker toolbar
240.         CGRect viewFrame = self.view.frame;
241.         viewFrame.size.height -= (self.pickerToolBar.frame.size.height);
242.
243.         // Shrink the table vertical size to make room for the date picker
244.         viewFrame.size.height -= (self.pickerViewRentOrMortgage.frame.size.height - 50);
245.         self.view.frame = viewFrame;
246.
247.         [UIView commitAnimations];
248.
249.         // Scroll the active picker button into view.
250.         CGRect rentFromPickerButtonRect = [rentFromPickerButton frame];
251.         [scrollExpensesView scrollRectToVisible:rentFromPickerButtonRect animated:YES];
252.     }

```

```

253. }
254.
255. // Show picker view rent or mortgage without slide animation
256. -(void)popUpWithoutSlidePickerViewRentOrMortgage
257. {
258.     // Add the picker toolbar
259.     [self.view.window addSubview: self.pickerToolBar];
260.     //Add to picker view
261.     [self.view.window addSubview: self.pickerViewRentOrMortgage];
262.
263.     // Size up the picker view to the screen and compute the start/end frame origin for the slide
    up animation
264.     //
265.     // Compute the start frame
266.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
267.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
268.     CGSize pickerSize = [self.pickerViewRentOrMortgage sizeThatFits:CGSizeZero];
269.     CGRect startRectPickerToolBar = CGRectMake(0.0,
270.                                                screenRect.origin.y +
    screenRect.size.height,
271.                                                pickerToolBarSize.width,
    pickerToolBarSize.height);
272.     CGRect startRectPicker = CGRectMake(0.0,
273.                                                screenRect.origin.y +
    screenRect.size.height,
274.                                                pickerSize.width, pickerSize.height);
275.     self.pickerToolBar.frame = startRectPickerToolBar;
276.     self.pickerViewRentOrMortgage.frame = startRectPicker;
277.
278.     // Compute the end frame
279.     CGRect pickerToolBarRect = CGRectMake(0.0,
280.                                                screenRect.origin.y +
    screenRect.size.height - (pickerToolBarSize.height + pickerSize.height),
281.                                                pickerToolBarSize.width,
    pickerToolBarSize.height);
282.
283.

```

```

284.         // Compute the end frame
285.         CGRect pickerRect = CGRectMake(0.0,
286.         pickerSize.height,
287.         screenRect.origin.y + screenRect.size.height -
288.         pickerSize.width,
289.         pickerSize.height);
290.         self.pickerToolBar.frame = pickerToolBarRect;
291.         self.pickerViewRentOrMortgage.frame = pickerRect;
292.     }
293.
294. // Show picker view rent from with slide animation
295. - (void)popUpSlidePickerViewRentFrom
296. {
297.     // If keyboard is active then release it
298.     if(activeField != nil)
299.         [activeField resignFirstResponder];
300.
301.     // Check if date picker is already on screen
302.     if (self.pickerViewRentFrom.superview == nil)
303.     {
304.         // Add the picker toolbar
305.         [self.view.window addSubview: self.pickerToolBar];
306.         //Add to picker view
307.         [self.view.window addSubview: self.pickerViewRentFrom];
308.
309.         // Size up the picker view to the screen and compute the start/end frame origin for the
310.         // slide up animation
311.         // Compute the start frame
312.         CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
313.         CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
314.         CGSize pickerSize = [self.pickerViewRentFrom sizeThatFits:CGSizeZero];
315.         CGRect startRectPickerToolbar = CGRectMake(0.0,
316.         screenRect.size.height,
         screenRect.origin.y +

```

```

317.                                     pickerToolBarSize.width,
    pickerToolBarSize.height);
318.         CGRect startRectPicker = CGRectMake(0.0,
319.         screenRect.size.height,                                     screenRect.origin.y +
    pickerSize.height);                                           pickerSize.width,
320.                                     screenRect.origin.y +
    pickerSize.height);
321.         self.pickerToolBar.frame = startRectPickerToolBar;
322.         self.pickerViewRentFrom.frame = startRectPicker;
323.
324.         // Compute the end frame
325.         CGRect pickerToolBarRect = CGRectMake(0.0,
326.         screenRect.size.height - (pickerToolBarSize.height + pickerSize.height),
    screenRect.origin.y +
    pickerToolBarSize.width,
    pickerToolBarSize.height);
327.                                     pickerToolBarSize.width,
    pickerToolBarSize.height);
328.
329.
330.         // Compute the end frame
331.         CGRect pickerRect = CGRectMake(0.0,
332.         screenRect.origin.y + screenRect.size.height
    - pickerSize.height,
    pickerSize.width,
    pickerSize.height);
333.
334.
335.
336.         // Start the slide up animation
337.         [UIView beginAnimations:nil context:NULL];
338.         [UIView setAnimationDuration:0.3];
339.
340.         // Need to perform some post operations after the animation is complete
341.         [UIView setAnimationDelegate:self];
342.
343.         self.pickerToolBar.frame = pickerToolBarRect;
344.         self.pickerViewRentFrom.frame = pickerRect;
345.
346.         // Shrink the table vertical size to make room for the picker toolbar
347.         CGRect viewFrame = self.view.frame;

```

```

348.         viewFrame.size.height -= (self.pickerToolBar.frame.size.height);
349.
350.         // Shrink the table vertical size to make room for the date picker
351.         viewFrame.size.height -= (self.pickerViewRentFrom.frame.size.height - 50);
352.         self.view.frame = viewFrame;
353.
354.         [UIView commitAnimations];
355.
356.         // Scroll the active picker button into view.
357.         CGRect rentFromPickerButtonRect = [rentFromPickerButton frame];
358.         [scrollExpensesView scrollRectToVisible:rentFromPickerButtonRect animated:YES];
359.     }
360. }
361.
362. // Show picker view rent from without slide animation
363. -(void)popUpWithoutSlidePickerViewRentFrom
364. {
365.     // Add the picker toolbar
366.     [self.view.window addSubview: self.pickerToolBar];
367.     //Add to picker view
368.     [self.view.window addSubview: self.pickerViewRentFrom];
369.
370.     // Size up the picker view to the screen and compute the start/end frame origin for the slide
    up animation
371.     //
372.     // Compute the start frame
373.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
374.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
375.     CGSize pickerSize = [self.pickerViewRentFrom sizeThatFits:CGSizeZero];
376.     CGRect startRectPickerToolBar = CGRectMake(0.0,
377.                                                screenRect.origin.y +
    screenRect.size.height,
378.                                                pickerToolBarSize.width,
    pickerToolBarSize.height);
379.     CGRect startRectPicker = CGRectMake(0.0,

```

```

380.                                     screenRect.origin.y +
    screenRect.size.height,           pickerSize.width, pickerSize.height);
381.
382.     self.pickerToolBar.frame = startRectPickerToolBar;
383.     self.pickerViewRentFrom.frame = startRectPicker;
384.
385.     // Compute the end frame
386.     CGRect pickerToolBarRect = CGRectMake(0.0,
387.                                     screenRect.origin.y +
    screenRect.size.height - (pickerToolBarSize.height + pickerSize.height),
388.                                     pickerToolBarSize.width,
389.                                     pickerToolBarSize.height);
390.
391.     // Compute the end frame
392.     CGRect pickerRect = CGRectMake(0.0,
393.                                     screenRect.origin.y + screenRect.size.height -
    pickerSize.height,
394.                                     pickerSize.width,
395.                                     pickerSize.height);
396.
397.     self.pickerToolBar.frame = pickerToolBarRect;
398.     self.pickerViewRentFrom.frame = pickerRect;
399. }
400.
401. - (IBAction) rentOrMortgagePickerButtonPressed:(id)sender{
402.
403.     // Change the button label title to 'Select..'
404.     [rentOrMorgagePickerButton setTitle:@"          Select..          "
    forState:UIControlStateNormal];
405.     rentOrMortgageFlag = YES;
406.
407.     // If picker view is active then release it
408.     if(self.pickerViewRentFrom.superview != nil){
409.         // Remove pickerViewRentFrom from superview
410.         [self.pickerViewRentFrom removeFromSuperview];
411.

```

```

412.         [self popUpWithoutSlidePickerViewRentOrMortgage];
413.
414.         // Report the selection to the UI Button
415.         rentFromPickerButton.titleLabel.text = [pickerViewRentFromArray
    objectAtIndex:[pickerViewRentFrom selectedRowInComponent:0]];
416.     }
417.     else {
418.         [self popUpSlidePickerViewRentOrMortgage];
419.     }
420. }
421.
422. - (IBAction) rentFromPickerButtonPressed:(id)sender{
423.
424.     rentOrMortgageFlag = NO;
425.
426.     // If picker view is active then release it
427.     if(self.pickerViewRentOrMortgage.superview != nil){
428.         // Remove pickerViewRentOrMortgage from superview
429.         [self.pickerViewRentOrMortgage removeFromSuperview];
430.
431.         [self popUpWithoutSlidePickerViewRentFrom];
432.
433.         // Report the selection to the UI Button
434.         rentOrMortgagePickerButton.titleLabel.text = [pickerViewRentOrMortgageArray
    objectAtIndex:[pickerViewRentOrMortgage selectedRowInComponent:0]];
435.     }
436.     else {
437.         [self popUpSlidePickerViewRentFrom];
438.     }
439. }
440.
441. + (void) resignPickerView{
442.     [self resignPickerView];
443. }
444.
445. - (void) slideDownRentOrMortgageDidStop

```

```
446. {
447.     // The rent or mortgage picker has finished sliding downwards, so remove it
448.     [self.pickerViewRentOrMortgage removeFromSuperview];
449. }
450.
451. - (void)slideDownRentFromDidStop
452. {
453.     // The rent from picker has finished sliding downwards, so remove it
454.     [self.pickerViewRentFrom removeFromSuperview];
455. }
456.
457. - (void) resignPickerViewRentOrMortgage{
458.
459.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
460.     CGRect endFrameToolBar = self.pickerViewRentOrMortgage.frame;
461.     CGRect endFramePicker = self.pickerViewRentOrMortgage.frame;
462.     endFrameToolBar.origin.y = screenRect.origin.y + screenRect.size.height;
463.     endFramePicker.origin.y = screenRect.origin.y + screenRect.size.height;
464.
465.     // Start the slide down animation
466.     [UIView beginAnimations:nil context:NULL];
467.     [UIView setAnimationDuration:0.3];
468.
469.     // Need to perform some post operations after the animation is complete
470.     [UIView setAnimationDelegate:self];
471.     [UIView setAnimationDidStopSelector:@selector(slideDownRentOrMortgageDidStop)];
472.
473.     self.pickerToolBar.frame = endFrameToolBar;
474.     self.pickerViewRentOrMortgage.frame = endFramePicker;
475.     [UIView commitAnimations];
476.
477.     // Grow the table back again in vertical size to make room for the date picker
478.     CGRect viewFrame = self.view.frame;
479.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
480.     CGSize pickerSize = [self.pickerViewRentOrMortgage sizeThatFits:CGSizeZero];
```



```

481.         viewFrame.size.height += (self.pickerToolBar.frame.size.height - (pickerSize.height -
pickerToolBarSize.height));
482.         viewFrame.size.height += (self.pickerViewRentOrMortgage.frame.size.height - 50);
483.         self.view.frame = viewFrame;
484.
485.         // Report the selection to the UI Button
486.         rentOrMortgagePickerButton.titleLabel.text = [ickerViewRentOrMortgageArray
objectAtIndex:[ickerViewRentOrMortgage selectedRowInComponent:0]];
487.     }
488.
489. - (void) resignPickerViewRentFrom{
490.
491.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
492.     CGRect endFrameToolBar = self.pickerViewRentFrom.frame;
493.     CGRect endFramePicker = self.pickerViewRentFrom.frame;
494.     endFrameToolBar.origin.y = screenRect.origin.y + screenRect.size.height;
495.     endFramePicker.origin.y = screenRect.origin.y + screenRect.size.height;
496.
497.     // Start the slide down animation
498.     [UIView beginAnimations:nil context:NULL];
499.     [UIView setAnimationDuration:0.3];
500.
501.     // Need to perform some post operations after the animation is complete
502.     [UIView setAnimationDelegate:self];
503.     [UIView setAnimationDidStopSelector:@selector(slideDownRentFromDidStop)];
504.
505.     self.pickerToolBar.frame = endFrameToolBar;
506.     self.pickerViewRentFrom.frame = endFramePicker;
507.     [UIView commitAnimations];
508.
509.     // Grow the table back again in vertical size to make room for the rent from picker view
510.     CGRect viewFrame = self.view.frame;
511.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
512.     CGSize pickerSize = [self.pickerViewRentFrom sizeThatFits:CGSizeZero];
513.     viewFrame.size.height += (self.pickerToolBar.frame.size.height - (pickerSize.height -
pickerToolBarSize.height));

```

```

514.         viewFrame.size.height += (self.pickerViewRentFrom.frame.size.height - 50);
515.         self.view.frame = viewFrame;
516.
517.         // Report the selection to the UI Button
518.         rentFromPickerButton.titleLabel.text = [ickerViewRentFromArray
objectAtIndex:[ickerViewRentFrom selectedRowInComponent:0]];
519.     }
520.
521. - (IBAction)doneAction:(id)sender{
522.
523.     // Resign the active picker view
524.     if (self.pickerViewRentOrMortgage.superview != nil)
525.         [self resignPickerViewRentOrMortgage];
526.     else
527.         [self resignPickerViewRentFrom];
528. }
529.
530. - (void) addSubRentOrMortgageInfo{
531.
532.     // Add the additional Rent or Mortgage Informations
533.     if ([[rentOrMortgagePickerButton titleLabel] text] isEqualToString:@"Rent"]) {
534.         interestReliefLabel.hidden = YES;
535.         interestReliefSwitch.hidden = YES;
536.         interestReliefHelpButton.hidden = YES;
537.
538.         rentFromLabel.hidden = NO;
539.         blankImageView.hidden = YES;
540.         pickerDownArrowImageView.hidden = NO;
541.         rentFromPickerButton.hidden = NO;
542.         rentFromHelpButton.hidden = NO;
543.
544.         // Report the selection to the UI Button
545.         rentFromPickerButton.titleLabel.text = [ickerViewRentFromArray
objectAtIndex:[ickerViewRentFrom selectedRowInComponent:0]];
546.
547.         annualRentLabel.hidden = NO;

```

```
548.         annualRentEuroLabel.hidden = NO;
549.         annualRentTextField.hidden = NO;
550.         annualRentHelpButton.hidden = NO;
551.
552.         redOakManImageView.hidden = YES;
553.     }
554.
555.     else if( [[[rentOrMorgagePickerButton titleLabel] text] isEqualToString:@"Mortgage"]){
556.         rentFromLabel.hidden = YES;
557.         blankImageView.hidden = YES;
558.         pickerDownArrowImageView.hidden = YES;
559.         rentFromPickerButton.hidden = YES;
560.         rentFromHelpButton.hidden = YES;
561.
562.         annualRentLabel.hidden = YES;
563.         annualRentEuroLabel.hidden = YES;
564.         annualRentTextField.hidden = YES;
565.         annualRentHelpButton.hidden = YES;
566.
567.         interestReliefLabel.hidden = NO;
568.         interestReliefSwitch.hidden = NO;
569.         interestReliefHelpButton.hidden = NO;
570.
571.         redOakManImageView.hidden = YES;
572.     }
573.
574.     else {
575.         rentFromLabel.hidden = YES;
576.         pickerDownArrowImageView.hidden = YES;
577.         rentFromPickerButton.hidden = YES;
578.         rentFromHelpButton.hidden = YES;
579.
580.         annualRentLabel.hidden = YES;
581.         annualRentEuroLabel.hidden = YES;
582.         annualRentTextField.hidden = YES;
583.         annualRentHelpButton.hidden = YES;
```

```
584.
585.     interestReliefLabel.hidden = YES;
586.     interestReliefSwitch.hidden = YES;
587.     interestReliefHelpButton.hidden = YES;
588.
589.     redOakManImageView.hidden = NO;
590.     }
591.
592. }
593.
594. - (void)textFieldDidBeginEditing:(UITextField *)textField
595. {
596.     // If picker view is active then release it
597.     if(self.pickerViewRentOrMortgage.superview != nil){
598.         [self resignPickerViewRentOrMortgage];
599.     }
600.     else if (self.pickerViewRentFrom.superview != nil) {
601.         [self resignPickerViewRentFrom];
602.     }
603.
604.     activeField = textField;
605. }
606.
607. - (void)textFieldDidEndEditing:(UITextField *)textField
608. {
609.     // When editing is finished, seperate number its fractional parts
610.     [textField setText:[self commaSeparateThousands: [[textField text] intValue]]];
611.
612.     activeField = nil;
613. }
614.
615. - (void)viewWillAppear:(BOOL)animated{
616.     [super viewWillAppear:animated];
617.     // Registering for keyboard events
618.     [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector (keyboardDidShow:)
```

```
619.     name:UIKeyboardDidShowNotification object:nil];
620.         [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector (keyboardWillHide:)
621.     name:UIKeyboardWillHideNotification object:nil];
622.
623.         // Initially the keyboard is hidden, so reset our variable
624.         keyboardVisible = NO;
625.
626.         // Registering for text field events
627.         [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(limitTextField:)
628.     name:@"UITextFieldTextDidChangeNotification" object:activeField];
629. }
630.
631. - (void)viewWillDisappear:(BOOL)animated{
632.     // Unregistering for keyboard events
633.     [[NSNotificationCenter defaultCenter] removeObserver:self];
634.
635.     // Unregistering for text field events
636.     [[NSNotificationCenter defaultCenter] removeObserver:self
637.     name:@"UITextFieldTextDidChangeNotification" object:activeField];
638. }
639. - (void)keyboardDidShow:(NSNotification *)notif {
640.
641.     if(activeField == annualRentTextField){
642.         if (keyboardVisible) {
643.             // Keyboard is already visible. Ignoring notification
644.             return;
645.         }
646.
647.         //The keyboard wasn't visible before
648.         // Resizing smaller for keyboard
649.
650.         //Get the size of the keyboard
```

```
651.     NSDictionary *info = [notif userInfo];
652.     NSValue *aValue = [info objectForKey:UIKeyboardBoundsUserInfoKey];
653.     CGSize keyboardSize = [aValue CGRectValue].size;
654.
655.     //Resize the scroll view to make the room for the keyboard
656.     CGRect viewFrame = self.view.frame;
657.     viewFrame.size.height -= (keyboardSize.height);
658.     scrollExpensesView.frame = viewFrame;
659.
660.     // Scroll the active text field into view.
661.     CGRect textFieldRect = [activeField frame];
662.     [scrollExpensesView scrollRectToVisible:textFieldRect animated:YES];
663.
664.     keyboardVisible = YES;
665. }
666. }
667.
668. - (void)keyboardWillHide:(NSNotification *)notif {
669.
670.     if(!keyboardVisible){
671.         // Keyboard already hidden. Ignoring notification
672.         return;
673.     }
674.
675.     // The keyboard was visible
676.     // Resizing bigger with no keyboard
677.
678.     //Reset the height of the scroll view to its original value
679.     CGRect viewFrame = self.view.frame;
680.     scrollExpensesView.frame = viewFrame;
681.
682.     keyboardVisible = NO;
683. }
684.
685. -(void) resignKeyboard
686. {
```

```

687.     if(!keyboardVisible){
688.         // Keyboard already hidden. Ignoring notification
689.         return;
690.     }
691.
692.     // The keyboard was visible
693.     // Resizing bigger with no keyboard
694.
695.     CGRect viewFrame = self.view.frame;
696.     scrollExpensesView.frame = viewFrame;
697.
698.     keyboardVisible = NO;
699. }
700.
701. +(void) resignKeyboard
702. {
703.     [self resignKeyboard];
704. }
705.
706. -(IBAction) interestReliefSwitchValueChanged
707. {
708.     // If picker view rent or mortgage is active, resign the picker
709.     if (self.pickerViewRentOrMortgage.superview != nil)
710.         [self resignPickerViewRentOrMortgage];
711. }
712.
713. // Thousand Separator
714. -(NSString*) commaSeparateThousands: (int) number
715. {
716.     NSString *result = @"";
717.     while( number > 0 )
718.     {
719.         int removedDigits = number % 1000;
720.         if( result != nil && [result length] > 0 )
721.         {
722.             if( number > 999 )

```

```

723.             result = [NSString stringWithFormat:@"%%.3i,%@", removedDigits, result];
724.         else
725.             result = [NSString stringWithFormat:@"%i,%@", removedDigits, result];
726.     }
727.     else
728.     {
729.         if( number > 999 )
730.             result = [NSString stringWithFormat:@"%%.3i", removedDigits];
731.         else
732.             result = [NSString stringWithFormat:@"%i", removedDigits];
733.     }
734.     number = number/1000;
735. }
736.
737.     return result;
738. }
739.
740. // Limit the text field input
741. - (void)limitTextField:(NSNotification *)notif {
742.     int limit = 9;
743.     if ([[activeField text] length] > limit) {
744.         [activeField setText:[activeField text] substringToIndex:limit]];
745.     }
746. }
747.
748. /*
749. // The designated initializer.  Override if you create the controller programmatically and want to
    perform customization that is not appropriate for viewDidLoad.
750. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
751.     if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
752.         // Custom initialization
753.     }
754.     return self;
755. }
756. */
757.

```



```
758. /*
759. // Implement loadView to create a view hierarchy programmatically, without using a nib.
760. - (void)loadView {
761. }
762. */
763.
764. /*
765. // Override to allow orientations other than the default portrait orientation.
766. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
767.     // Return YES for supported orientations
768.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
769. }
770. */
771.
772. - (void)didReceiveMemoryWarning {
773.     // Releases the view if it doesn't have a superview.
774.     [super didReceiveMemoryWarning];
775.
776.     // Release any cached data, images, etc that aren't in use.
777. }
778.
779. - (void)viewDidUnload {
780.     self.pickerToolBar = nil;
781.     // Release any retained subviews of the main view.
782.     // e.g. self.myOutlet = nil;
783. }
784.
785.
786. - (void)dealloc {
787.     [scrollExpensesView release];
788.     [pickerViewRentOrMortgage release];
789.     [pickerToolBar release];
790.     [pickerViewRentOrMortgageArray release];
791.     [pickerViewRentFromArray release];
792.     [rentOrMorgagePickerButton release];
793.     [rentFromPickerButton release];
```

```
794.         [allowableMedicalTextField release];
795.         [allowableDentalTextField release];
796.         [binChargesTextField release];
797.         [annualRentTextField release];
798.         [interestReliefSwitch release];
799.         [rentFromLabel release];
800.         [blankImageView release];
801.         [pickerDownArrowImageView release];
802.         [rentFromHelpButton release];
803.         [annualRentLabel release];
804.         [annualRentEuroLabel release];
805.         [annualRentHelpButton release];
806.         [interestReliefLabel release];
807.         [interestReliefHelpButton release];
808.         [redOakManImageView release];
809.         [pickerViewRentFrom release];
810.     [super dealloc];
811. }
812.
813. @end
```

2.9. SpouseViewController.h

```
1. //
2. // SpouseViewController.h
3. // TaxCalc2009
4. //
5. // Created by C00115609 on 24/02/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8.
9. #import <UIKit/UIKit.h>
10.
11. @interface SpouseViewController : UIViewController {
12.
```

```

13.      IBOutlet UIScrollView *scrollSpouseView;
14.      IBOutlet UIButton *dateOfBirthPickerButton;
15.      IBOutlet UIToolbar *pickerToolBar;          // this toolbar appears only when the date picker is
open
16.      IBOutlet UIDatePicker *datePicker;
17.
18.      IBOutlet UITextField *spouseAnnualSalaryTextField;
19.      IBOutlet UITextField *spouseTaxOnPayTextField;
20.      IBOutlet UISwitch *spouseTradeUnionSwitch;
21.
22.      NSDateFormatter *dateFormatter;
23.      UITextField *activeField;
24.  }
25.
26.  @property (nonatomic, retain) IBOutlet UIScrollView *scrollSpouseView;
27.  @property (nonatomic, retain) IBOutlet UIDatePicker *datePicker;
28.  @property (nonatomic, retain) IBOutlet UIToolbar *pickerToolBar;
29.  @property (nonatomic, retain) NSDateFormatter *dateFormatter;
30.
31.  @property (nonatomic, retain) IBOutlet UIButton *dateOfBirthPickerButton;
32.  @property (nonatomic, retain) IBOutlet UITextField *spouseAnnualSalaryTextField;
33.  @property (nonatomic, retain) IBOutlet UITextField *spouseTaxOnPayTextField;
34.  @property (nonatomic, retain) IBOutlet UISwitch *spouseTradeUnionSwitch;
35.
36.  - (IBAction) dateOfBirthHelpButtonPressed;
37.  - (IBAction) anualSalaryBeforeTaxHelpButtonPressed;
38.  - (IBAction) TaxOnPayHelpButtonPressed;
39.  - (IBAction) MemberTradeUnionHelpButtonPressed;
40.
41.  - (IBAction) textFieldDoneEditing:(id) sender;
42.  - (IBAction) dateOfBirthPickerButtonPressed:(id) sender;
43.  - (IBAction) doneAction:(id) sender; // when the done button is clicked
44.  - (IBAction) dateAction:(id) sender; // when the user has changed the date picker values (m/d/y)
45.  - (IBAction) tradeUnionMemberSegControlValueChanged;
46.
47.  - (void)textFieldDidBeginEditing:(UITextField *)textField;

```

```
48. - (void)textFieldDidEndEditing:(UITextField *)textField;
49.
50. - (void) resignPickerView;
51. + (void) resignPickerView;
52. - (void) limitTextField:(NSNotification *)notif;
53.
54. -(NSString*) commaSeparateThousands: (int) number;
55.
56. @end
```

2.10. SpouseViewController.m

```
1. //
2. // SpouseViewController.m
3. // TaxCalc2009
4. //
5. // Created by C00115609 on 24/02/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8. /*****
9.             Class Description
10.    This class is responsible for controlling the "Spouse" view's functionalities.
11.    *****/
12.
13. #import "SpouseViewController.h"
14.
15. @implementation SpouseViewController
16.
17. @synthesize scrollSpouseView, datePicker, dateFormatter, pickerToolBar;
18. @synthesize spouseAnnualSalaryTextField, spouseTaxOnPayTextField, spouseTradeUnionSwitch,
    dateOfBirthPickerButton;
19.
20. // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
21. - (void)viewDidLoad {
22.     [super viewDidLoad];
```

```

23.
24.     scrollSpouseView.contentSize = CGSizeMake(320, 461);
25.
26.     self.dateFormatter = [[NSDateFormatter alloc] autorelease];
27.     [self.dateFormatter setDateStyle:NSDateFormatterShortStyle];
28.     [self.dateFormatter setTimeStyle:NSDateFormatterNoStyle];
29.
30.     // Change the Trade Union Switch (Slider) text (YES/NO instead ON/OFF)
31.     [(UILabel *)[[[[[spouseTradeUnionSwitch subviews] lastObject] subviews] objectAtIndex:2]
subviews] objectAtIndex:0] setText:@"YES"];
32.     [(UILabel *)[[[[[spouseTradeUnionSwitch subviews] lastObject] subviews] objectAtIndex:2]
subviews] objectAtIndex:1] setText:@"NO"];
33. }
34.
35. -(IBAction) dateOfBirthHelpButtonPressed
36. {
37.     UIAlertView *alert = [[UIAlertView alloc]
38.                             initWithTitle:@"Tips: Date of Birth"
39.                             message:@"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."
40.                             delegate:nil
41.                             cancelButtonTitle:@"OK"
42.                             otherButtonTitles:nil];
43.     [alert show];
44.     [alert release];
45. }
46.
47. -(IBAction) anualSalaryBeforeTaxHelpButtonPressed
48. {
49.     UIAlertView *alert = [[UIAlertView alloc]
50.                             initWithTitle:@"Tips: Annual Salary before Tax"
51.                             message:@"What's your spouses annual salary before tax and
after pension deductions. This is the gross pay from the P60 (or P45)"
52.                             delegate:nil
53.                             cancelButtonTitle:@"OK"
54.                             otherButtonTitles:nil];

```

```

55.     [alert show];
56.     [alert release];
57. }
58.
59. - (IBAction) TaxOnPayHelpButtonPressed
60. {
61.     UIAlertView *alert = [[UIAlertView alloc]
62.         initWithTitle:@"Tips: Tax on Salary"
63.         message:@"How much tax did your spouse pay on that salary?
        Don't include other deductions like PRSI or income levy here, just income tax."
64.         delegate:nil
65.         cancelButtonTitle:@"OK"
66.         otherButtonTitles:nil];
67.     [alert show];
68.     [alert release];
69. }
70.
71. - (IBAction) MemberTradeUnionHelpButtonPressed
72. {
73.     UIAlertView *alert = [[UIAlertView alloc]
74.         initWithTitle:@"Tips: Member of a Trade Union?"
75.         message:@"If your spouse is in a Trade Union you are entitled
        to claim a tax credit."
76.         delegate:nil
77.         cancelButtonTitle:@"OK"
78.         otherButtonTitles:nil];
79.     [alert show];
80.     [alert release];
81. }
82.
83. - (IBAction) dateOfBirthPickerButtonPressed:(id) sender
84. {
85.
86.     // If keyboard is active then release it
87.     if(activeField != nil)
88.         [activeField resignFirstResponder];

```

```

89.
90.     // Check if date picker is already on screen
91.     if (self.datePicker.superview == nil)
92.     {
93.         // Add the picker toolbar
94.         [self.view.window addSubview: self.pickerToolBar];
95.         //Add to picker view
96.         [self.view.window addSubview: self.datePicker];
97.
98.         // Size up the picker view to the screen and compute the start/end frame origin for the
    slide up animation
99.         //
100.        // Compute the start frame
101.        CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
102.        CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
103.        CGSize pickerSize = [self.datePicker sizeThatFits:CGSizeZero];
104.        CGRect startRectPickerToolBar = CGRectMake(0.0,
105.                                                    screenRect.origin.y +
    screenRect.size.height,
106.                                                    pickerToolBarSize.width,
    pickerToolBarSize.height);
107.        CGRect startRectPicker = CGRectMake(0.0,
108.                                                    screenRect.origin.y +
    screenRect.size.height,
109.                                                    pickerSize.width,
    pickerSize.height);
110.        self.pickerToolBar.frame = startRectPickerToolBar;
111.        self.datePicker.frame = startRectPicker;
112.
113.        // Compute the end frame
114.        CGRect pickerToolBarRect = CGRectMake(0.0,
115.                                                    screenRect.origin.y +
    screenRect.size.height - (pickerToolBarSize.height + pickerSize.height),
116.                                                    pickerToolBarSize.width,
    pickerToolBarSize.height);
117.
118.

```

```

119.         // Compute the end frame
120.         CGRect pickerRect = CGRectMake(0.0,
121.         - pickerSize.height,
122.         screenRect.origin.y + screenRect.size.height
123.         pickerSize.width,
124.         pickerSize.height);
125.
126.         // Start the slide up animation
127.         [UIView beginAnimations:nil context:NULL];
128.         [UIView setAnimationDuration:0.3];
129.
130.         // Need to perform some post operations after the animation is complete
131.         [UIView setAnimationDelegate:self];
132.
133.         self.pickerToolBar.frame = pickerToolBarRect;
134.         self.datePicker.frame = pickerRect;
135.
136.         // Shrink the table vertical size to make room for the picker toolbar
137.         CGRect viewFrame = self.view.frame;
138.         viewFrame.size.height -= (self.pickerToolBar.frame.size.height);
139.         self.view.frame = viewFrame;
140.
141.         // Shrink the table vertical size to make room for the date picker
142.         viewFrame.size.height -= (self.datePicker.frame.size.height - 50);
143.         self.view.frame = viewFrame;
144.
145.         [UIView commitAnimations];
146.     }
147.
148. // the date picker has finished sliding downwards, so remove it
149. - (void)slideDownDidStop
150. {
151.     [self.datePicker removeFromSuperview];
152. }
153.

```



```

154. // Set the picker button value when the user changes date picker value
155. - (IBAction)dateAction:(id)sender
156. {
157.     dateOfBirthPickerButton.titleLabel.text = [self.dateFormatter
    stringFromDate:self.datePicker.date];
158. }
159.
160. + (void) resignPickerView{
161.     [self resignPickerView];
162. }
163.
164. - (void) resignPickerView{
165.     CGRect screenRect = [[UIScreen mainScreen] applicationFrame];
166.     CGRect endFrameToolBar = self.datePicker.frame;
167.     CGRect endFramePicker = self.datePicker.frame;
168.     endFrameToolBar.origin.y = screenRect.origin.y + screenRect.size.height;
169.     endFramePicker.origin.y = screenRect.origin.y + screenRect.size.height;
170.
171.     // start the slide down animation
172.     [UIView beginAnimations:nil context:NULL];
173.     [UIView setAnimationDuration:0.3];
174.
175.     // we need to perform some post operations after the animation is complete
176.     [UIView setAnimationDelegate:self];
177.     [UIView setAnimationDidStopSelector:@selector(slideDownDidStop)];
178.
179.     self.pickerToolBar.frame = endFrameToolBar;
180.     self.datePicker.frame = endFramePicker;
181.     [UIView commitAnimations];
182.
183.     // grow the table back again in vertical size to make room for the date picker
184.     CGRect viewFrame = self.view.frame;
185.     CGSize pickerToolBarSize = [self.pickerToolBar sizeThatFits:CGSizeZero];
186.     CGSize pickerSize = [self.datePicker sizeThatFits:CGSizeZero];
187.     viewFrame.size.height += (self.pickerToolBar.frame.size.height - (pickerSize.height -
    pickerToolBarSize.height));

```

```
188.         viewFrame.size.height += (self.datePicker.frame.size.height - 50);
189.         self.view.frame = viewFrame;
190.
191.         // Report the selection to the UI Button
192.         dateOfBirthPickerButton.titleLabel.text = [self.dateFormatter
stringFromDate:self.datePicker.date];;
193.     }
194.
195. - (IBAction)doneAction:(id)sender{
196.     [self resignPickerView];
197. }
198.
199. - (void)textFieldDidBeginEditing:(UITextField *)textField
200. {
201.     activeField = textField;
202.
203.     // If picker view is active then release it
204.     if (self.datePicker.superview != nil) {
205.         [self resignPickerView];
206.     }
207. }
208.
209. - (void)textFieldDidEndEditing:(UITextField *)textField
210. {
211.     // When editing is finished, seperate number its fractional parts
212.     [textField setText:[self commaSeparateThousands: [[textField text] intValue]]];
213.
214.     activeField = nil;
215. }
216.
217. /* Resign the keyboard */
218. -(IBAction) textFieldDoneEditing:(id)sender{
219.     [sender resignFirstResponder];
220. }
221.
222. - (IBAction) tradeUnionMemberSegControlValueChanged{
```

```

223.
224.     // If picker view is active then release it
225.     if (self.datePicker.superview != nil) {
226.         [self resignPickerView];
227.     }
228.     // If keyboard is active then release it
229.     else if(activeField != nil)
230.         [activeField resignFirstResponder];
231. }
232.
233. // Thousand Separator
234. -(NSString*) commaSeparateThousands: (int) number
235. {
236.     NSString *result = @"";
237.     while( number > 0 )
238.     {
239.         int removedDigits = number % 1000;
240.         if( result != nil && [result length] > 0 )
241.         {
242.             if( number > 999 )
243.                 result = [NSString stringWithFormat:@"%%.3i,%@", removedDigits, result];
244.             else
245.                 result = [NSString stringWithFormat:@"%i,%@", removedDigits, result];
246.         }
247.         else
248.         {
249.             if( number > 999 )
250.                 result = [NSString stringWithFormat:@"%%.3i", removedDigits];
251.             else
252.                 result = [NSString stringWithFormat:@"%i", removedDigits];
253.         }
254.         number = number/1000;
255.     }
256.
257.     return result;
258. }

```

```

259.
260. - (void)viewWillAppear:(BOOL)animated{
261.     [super viewWillAppear:animated];
262.     // Registering for text field events
263.     [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(limitTextField:)
264.     name:@"UITextFieldTextDidChangeNotification" object:activeField];
265. }
266.
267. - (void)viewWillDisappear:(BOOL)animated{
268.     // Unregistering for text field events
269.     [[NSNotificationCenter defaultCenter] removeObserver:self
270.     name:@"UITextFieldTextDidChangeNotification" object:activeField];
271. }
272. // Limit the text field input
273. - (void)limitTextField:(NSNotification *)notif {
274.     int limit = 9;
275.     if ([[activeField text] length] > limit) {
276.         [activeField setText:[activeField text] substringToIndex:limit]];
277.     }
278. }
279.
280. /*
281. // The designated initializer. Override if you create the controller programmatically and want to
282. // perform customization that is not appropriate for viewDidLoad.
283. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
284.     if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
285.         // Custom initialization
286.     }
287.     return self;
288. }
289. */
290. /*
291. // Implement loadView to create a view hierarchy programmatically, without using a nib.

```

```
292. - (void)loadView {
293. }
294. */
295.
296. /*
297. // Override to allow orientations other than the default portrait orientation.
298. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
299.     // Return YES for supported orientations
300.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
301. }
302. */
303.
304. - (void)didReceiveMemoryWarning {
305.     // Releases the view if it doesn't have a superview.
306.     [super didReceiveMemoryWarning];
307.
308.     // Release any cached data, images, etc that aren't in use.
309. }
310.
311. - (void)viewDidUnload {
312.     self.pickerToolBar = nil;
313.     self.dateFormatter = nil;
314.     // Release any retained subviews of the main view.
315.     // e.g. self.myOutlet = nil;
316. }
317.
318.
319. - (void)dealloc {
320.     [scrollSpouseView release];
321.     [datePicker release];
322.     [dateFormatter release];
323.     [pickerToolBar release];
324.     [spouseAnnualSalaryTextField release];
325.     [spouseTaxOnPayTextField release];
326.     [spouseTradeUnionSwitch release];
327.     [dateOfBirthPickerButton release];
```

```
328.     [super dealloc];
329. }
330.
331. @end
```

2.11. RefundResultViewController.h

```
1. //
2. // RefundResultViewController.h
3. // View_Switch
4. //
5. // Created by C00115609 on 29/01/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8.
9. #import <UIKit/UIKit.h>
10.
11.
12. @interface RefundResultViewController : UIViewController {
13.
14.     IBOutlet UILabel *greenLabel;
15.     IBOutlet UILabel *warningLabel;
16.     IBOutlet UITextField *refundEstimateTextField;
17. }
18.
19. @property (nonatomic, retain) IBOutlet UILabel *greenLabel;
20. @property (nonatomic, retain) IBOutlet UILabel *warningLabel;
21. @property (nonatomic, retain) IBOutlet UITextField *refundEstimateTextField;
22.
23. - (IBAction) detailClosureButtonPressed:(id) sender;
24.
25. - (void)updateCounter:(NSTimer *)theTimer;
26.
27. @end
```

2.12. RefundResultViewController.m

```
1. //
2. // RefundResultViewController.m
3. // View_Switch
4. //
5. // Created by C00115609 on 29/01/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8. /*****
9.                                     Class Description
10.    This class is responsible for controlling the "Your Tax Refund" view's functionalities.
11.    *****/
12.
13. #import "RefundResultViewController.h"
14.
15. @implementation RefundResultViewController
16. @synthesize refundEstimateTextField, greenLabel, warningLabel;
17.
18. // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
19. - (void)viewDidLoad {
20.     [super viewDidLoad];
21.
22.     // Timer for refund result animation
23.     [NSTimer scheduledTimerWithTimeInterval:1.0f
24.                                     target:self
25.                                     selector:@selector(updateCounter:)
26.                                     userInfo:nil
27.                                     repeats:YES];
28. }
29.
30. // Shutdown the app and go to Red Oak Company's website
31. - (IBAction) detailClosureButtonPressed:(id)sender {
32.     [[UIApplication sharedApplication] openURL:[NSURL
33.     URLWithString:@"http://www.redoaktaxrefunds.ie"]];
33. }
```

```

34.
35. // Set the animation for the result of the tax refund
36. - (void)updateCounter:(NSTimer *)theTimer {
37.     static int count = 24;
38.
39.     refundEstimateTextField.font = [UIFont fontWithName:@"Helvetica-Bold" size:count];
40.
41.     if (count < 34) {
42.         count += 10;
43.     }
44.     else {
45.         count -= 10;
46.     }
47.
48.     refundEstimateTextField.font = [UIFont fontWithName:@"Helvetica-Bold" size:count];
49. }
50.
51. /*
52. // The designated initializer. Override if you create the controller programmatically and want to
perform customization that is not appropriate for viewDidLoad.
53. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
54.     if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
55.         // Custom initialization
56.     }
57.     return self;
58. }
59. */
60.
61. /*
62. // Implement loadView to create a view hierarchy programmatically, without using a nib.
63. - (void)loadView {
64. }
65. */
66.
67. /*
68. // Override to allow orientations other than the default portrait orientation.

```



```
69. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
70.     // Return YES for supported orientations
71.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
72. }
73. */
74.
75. - (void)didReceiveMemoryWarning {
76.     // Releases the view if it doesn't have a superview.
77.     [super didReceiveMemoryWarning];
78.
79.     // Release any cached data, images, etc that aren't in use.
80. }
81.
82. - (void)viewDidUnload {
83.     // Release any retained subviews of the main view.
84.     // e.g. self.myOutlet = nil;
85. }
86.
87.
88. - (void)dealloc {
89.     [refundEstimateTextField release];
90.     [greenLabel release];
91.     [warningLabel release];
92.     [super dealloc];
93. }
94.
95. @end
```

2.13. ApplyViewController.h

```
1. //
2. //  ApplyViewController.h
3. //  View_Switch
4. //
5. //  Created by C00115609 on 04/02/2010.
6. //  Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8.
9. #import <UIKit/UIKit.h>
10. #import <MessageUI/MessageUI.h>
11. #import <MessageUI/MFMailComposeViewController.h>
12.
13. @interface ApplyViewController : UIViewController <MFMailComposeViewControllerDelegate,
    UITextFieldDelegate>{
14.
15.     IBOutlet UIScrollView *scrollView;
16.     IBOutlet UITextField *firstNameTextField;
17.     IBOutlet UITextField *lastNameTextField;
18.     IBOutlet UITextField *emailAddressTextField;
19.     IBOutlet UITextField *confirmEmailTextField;
20.     IBOutlet UITextField *phoneTextField;
21.     IBOutlet UITextField *companyTextField;
22.
23.     NSString *strCustomerDetails;
24.     UITextField *activeField;
25.     BOOL keyboardVisible;
26. }
27.
28. @property (nonatomic, retain) IBOutlet UIScrollView *scrollView;
29.
30. - (IBAction) submitButtonPressed:(id) sender;
31. - (IBAction) textFieldDoneEditing:(id) sender;
32.
33. - (void) keyboardDidShow:(NSNotification *)notif;
```

```
34. - (void) keyboardWillHide:(NSNotification *)notif;
35. - (void) textFieldDidBeginEditing:(UITextField *)textField;
36. - (void) textFieldDidEndEditing:(UITextField *)textField;
37.
38. - (void) launchMailAppOnDevice:(NSString *)strBody;
39. - (void) displayComposerSheet:(NSString *)strBody;
40.
41. @end
```

2.14. ApplyViewController.m

```
1. //
2. // ApplyViewController.m
3. // View_Switch
4. //
5. // Created by C00115609 on 04/02/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8. /*****
9.                                     Class Description
10.    This class is responsible for controlling the "Contact Us" view's functionalities.
11.    *****/
12.
13. #import "ApplyViewController.h"
14.
15. @implementation ApplyViewController
16.
17. @synthesize scrollView;
18.
19. // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
20. - (void)viewDidLoad {
21.     [super viewDidLoad];
22.     scrollView.contentSize = CGSizeMake(320, 411);
23. }
24.
```

```
25.  /* Resign the keyboard */
26.  -(IBAction) textFieldDoneEditing:(id)sender {
27.      [sender resignFirstResponder];
28.  }
29.
30.  - (void)textFieldDidBeginEditing:(UITextField *)textField
31.  {
32.      activeField = textField;
33.  }
34.
35.  - (void)textFieldDidEndEditing:(UITextField *)textField
36.  {
37.      activeField = nil;
38.  }
39.
40.  - (void)viewWillAppear:(BOOL)animated{
41.      [super viewWillAppear:animated];
42.      // Registering for keyboard events
43.      [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector (keyboardDidShow:)
44.      name:UIKeyboardDidShowNotification object:nil];
45.      [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector (keyboardWillHide:)
46.      name:UIKeyboardWillHideNotification object:nil];
47.      // Initially the keyboard is hidden, so reset our variable
48.      keyboardVisible = NO;
49.  }
50.
51.  - (void)viewWillDisappear:(BOOL)animated {
52.      // Unregistering for keyboard events
53.      [[NSNotificationCenter defaultCenter] removeObserver:self];
54.  }
55.
56.  - (void)keyboardDidShow:(NSNotification *)notif {
57.      if (keyboardVisible) {
58.          // Keyboard is already visible. Ignoring notification
```

```

59.         return;
60.     }
61.
62.     //The keyboard wasn't visible before
63.     // Resizing smaller for keyboard
64.
65.     //Get the size of the keyboard
66.     NSDictionary *info = [notif userInfo];
67.     NSValue *aValue = [info objectForKey:UIKeyboardBoundsUserInfoKey];
68.     CGSize keyboardSize = [aValue CGRectValue].size;
69.
70.     //Resize the scroll view to make the room for the keyboard
71.     CGRect viewFrame = self.view.frame;
72.     viewFrame.size.height -= (keyboardSize.height - 49);
73.     scrollView.frame = viewFrame;
74.
75.     // Scroll the active text field into view.
76.     CGRect textFieldRect = [activeField frame];
77.     [scrollView scrollRectToVisible:textFieldRect animated:YES];
78.
79.     keyboardVisible = YES;
80. }
81.
82. - (void)keyboardWillHide:(NSNotification *)notif {
83.     if(!keyboardVisible){
84.         // Keyboard already hidden. Ignoring notification
85.         return;
86.     }
87.
88.     // The keyboard was visible
89.     // Resizing bigger with no keyboard
90.
91.     //Get the size of the keyboard
92.     NSDictionary *info = [notif userInfo];
93.     NSValue *aValue = [info objectForKey:UIKeyboardBoundsUserInfoKey];
94.     CGSize keyboardSize = [aValue CGRectValue].size;

```

```
95.
96.     //Reset the height of the scroll view to its original value
97.     CGRect viewFrame = self.view.frame;
98.     viewFrame.size.height += (keyboardSize.height + 49);
99.     scrollView.frame = viewFrame;
100.
101.     keyboardVisible = NO;
102. }
103.
104. -(IBAction) submitButtonPressed:(id)sender
105. {
106.     // Generate the customer details string
107.     NSString strCustomerDetails = [NSString stringWithFormat:@"\nFirst Name: %@\nLast Name: %@\nEmail
Address: %@\nConfirm Email: %@\nPhone: %@\nCompany: %@",
108.     firstNameTextField.text, lastNameTextField.text, emailAddressTextField.text,
109.     confirmEmailTextField.text, phoneTextField.text, companyTextField.text];
110.
111.     Class mailClass = (NSClassFromString(@"MFMailComposeViewController"));
112.     if (mailClass != nil)
113.     {
114.         // We must always check whether the current device is configured for sending emails
115.         if ([mailClass canSendMail])
116.         {
117.             [self displayComposerSheet:strCustomerDetails];
118.         }
119.         else
120.         {
121.             [self launchMailAppOnDevice:strCustomerDetails];
122.         }
123.     }
124.     else
125.     {
126.         [self launchMailAppOnDevice:strCustomerDetails];
127.     }
```

```
128. }
129.
130. // Launches the Mail application on the device.
131. -(void)launchMailAppOnDevice:(NSString *)strBody
132. {
133.     NSString *recipients = @"mailto:john.oconnor@redoaktaxrefunds.ie?cc=&subject=Customer Details";
134.     NSString *body = [NSString stringWithFormat:@"%&body=%@", strBody];
135.
136.     NSString *email = [NSString stringWithFormat:@"%%%@", recipients, body];
137.     email = [email stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
138.
139.     [[UIApplication sharedApplication] openURL:[NSURL URLWithString:email]];
140. }
141.
142. #pragma mark -
143. #pragma mark Compose Mail
144. // Displays an email composition interface inside the application. Populates all the Mail fields.
145. -(void)displayComposerSheet:(NSString *)strBody
146. {
147.     MFMailComposeViewController *picker = [[MFMailComposeViewController alloc] init];
148.     picker.mailComposeDelegate = self;
149.
150.     [picker setSubject:@"Customer Details"];
151.
152.     // Set up recipients
153.     NSArray *toRecipients = [NSArray arrayWithObject:@"john.oconnor@redoaktaxrefunds.ie"];
154.
155.     [picker setToRecipients:toRecipients];
156.
157.     // Fill out the email body text
158.     NSString *emailBody = strBody;
159.     [picker setMessageBody:emailBody isHTML:NO];
160.
161.     [self presentModalViewController:picker animated:YES];
162.     [picker release];
163. }
```

```

164.
165. // Dismisses the email composition interface when users tap Cancel or Send. Proceeds to update the
    message field with the result of the operation.
166. - (void)mailComposeController:(MFMailComposeViewController*)controller
    didFinishWithResult:(MFMailComposeResult)result error:(NSError*)error
167. {
168.     UIAlertView *alertSent = [[UIAlertView alloc]
169.                               initWithTitle:@"Thank you for your Email"
170.                               message:@"Your details have been sent successfully. Our experts
    will contact with you shortly."
171.                               delegate:nil
172.                               cancelButtonTitle:@"OK"
173.                               otherButtonTitles:nil];
174.
175.     UIAlertView *alertFailed = [[UIAlertView alloc]
176.                                 initWithTitle:@"Sorry, Email Failed"
177.                                 message:@"Your details could not be sent. Please check your
    internet connection and try again."
178.                                 delegate:nil
179.                                 cancelButtonTitle:@"OK"
180.                                 otherButtonTitles:nil];
181.
182.     // Notifies users about errors associated with the interface
183.     switch (result)
184.     {
185.         case MFMailComposeResultCancelled:
186.             //message.text = @"Result: canceled";
187.             break;
188.         case MFMailComposeResultSaved:
189.             //message.text = @"Result: saved";
190.             break;
191.         case MFMailComposeResultSent:
192.             [alertSent show];
193.             [alertSent release];
194.             break;
195.         case MFMailComposeResultFailed:

```



```
196.         [alertViewFailed show];
197.         [alertViewFailed release];
198.         break;
199.     default:
200.         //message.text = @"Result: not sent";
201.         break;
202.     }
203.     [self dismissModalViewControllerAnimated:YES];
204. }
205.
206. /*
207. // The designated initializer. Override if you create the controller programmatically and want to
    perform customization that is not appropriate for viewDidLoad.
208. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
209.     if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
210.         // Custom initialization
211.     }
212.     return self;
213. }
214. */
215.
216. /*
217. // Implement loadView to create a view hierarchy programmatically, without using a nib.
218. - (void)loadView {
219. }
220. */
221.
222. /*
223. // Override to allow orientations other than the default portrait orientation.
224. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
225.     // Return YES for supported orientations
226.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
227. }
228. */
229.
230. - (void)didReceiveMemoryWarning {
```

```
231.         // Releases the view if it doesn't have a superview.
232.         [super didReceiveMemoryWarning];
233.
234.         // Release any cached data, images, etc that aren't in use.
235.     }
236.
237. - (void)viewDidUnload {
238.     // Release any retained subviews of the main view.
239.     // e.g. self.myOutlet = nil;
240. }
241.
242.
243. - (void)dealloc {
244.     [scrollView release];
245.     [super dealloc];
246. }
247.
248. @end
```

2.15. View_SwitchAppDelegate.h

```
1. //
2. //  View_SwitchAppDelegate.h
3. //  View_Switch
4. //
5. //  Created by C00115609 on 28/01/2010.
6. //  Copyright __Tuna Erdurmaz__ 2010. All rights reserved.
7. //
8.
9. #import <UIKit/UIKit.h>
10.
11. @interface View_SwitchAppDelegate : NSObject <UIApplicationDelegate> {
12.     UIWindow *window;
13.     IBOutlet UITabBarController *rootController;
14. }
15.
16. @property (nonatomic, retain) IBOutlet UIWindow *window;
17. @property (nonatomic, retain) IBOutlet UITabBarController *rootController;
18.
19. @end
```

2.16. View_SwitchAppDelegate.m

```
1. //
2. //  View_SwitchAppDelegate.m
3. //  View_Switch
4. //
5. //  Created by C00115609 on 28/01/2010.
6. //  Copyright __Tuna Erdurmaz__ 2010. All rights reserved.
7. //
8. /*****
9.
10.          Class Description
11.  This class is responsible for controlling application launch and termination, low-memory warnings,
12.  the opening of URL resources, changes in status-bar orientation, and other system events.
13.  The UIApplicationDelegate protocol in header file of this class declares methods that are implemented
14.  by the delegate of the singleton UIApplication object.
15.  *****/
16. #import "View_SwitchAppDelegate.h"
17.
18. @implementation View_SwitchAppDelegate
19.
20. @synthesize window;
21. @synthesize rootController;
22.
23. - (void)applicationDidFinishLaunching:(UIApplication *)application {
24.
25.     // Override point for customization after application launch
26.     [window addSubview:rootController.view];
27.     [window makeKeyAndVisible];
28. }
29.
30. - (void)dealloc {
31.     [window release];
32.     [rootController release];
33.     [super dealloc];
34. }
```

```
33.  }
34.
35.  @end
```

2.17. ConstantsTax.h

```
1.  //
2.  //  ConstantsTax.h
3.  //  TaxCalc2009
4.  //
5.  //  Created by C00115609 on 04/03/2010.
6.  //  Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7.  //
8.
9.  #import <Foundation/Foundation.h>
10. extern      NSInteger      const PersonalSingleTaxCredit;
11. extern      NSInteger      const PersonalSingleTaxBandMax;
12.
13. extern      NSInteger      const PersonalMarriedTaxCredit;
14. extern      NSInteger      const PersonalMarriedTaxBandMax;
15. extern      NSInteger      const PersonalSpouseTaxBandMax;
16. extern      NSInteger      const PersonalSpouseTaxCredit;
17.
18. extern      NSInteger      const PersonalSingleParentTaxCredit;
19. extern      NSInteger      const PersonalSingleParentTaxBandMax;
20.
21. extern      NSInteger      const HomeCarerTaxCredit;
22. extern      NSInteger      const HomeCarerSpouseAnnualSalaryMax;
23.
24. extern      NSInteger      const BinChargesMax;
25. extern      NSInteger      const TradeUnionCredit;
26. extern      NSInteger      const rentReliefSingleU65Max;
27. extern      NSInteger      const rentReliefSingleO65Max;
28. extern      NSInteger      const rentReliefMarriedU65Max;
29. extern      NSInteger      const rentReliefMarriedO65Max;
```

```

30. extern    NSInteger    const taxYear;
31.
32. extern    float        const TaxRateMax;
33. extern    float        const TaxRateMin;
34. extern    float        const TaxRateBin;
35. extern    float        const TaxRateRentRelief;
36. extern    float        const TaxRateBin;
37.
38.
39. @interface ConstantsTax : NSObject {
40.
41. }
42.
43. @end

```

2.18. ConstantsTax.m

```

1. //
2. //  ConstantsTax.m
3. //  TaxCalc2009
4. //
5. //  Created by C00115609 on 04/03/2010.
6. //  Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8. /*****
9.
10.          Class Description
11.  This class contains all the constant figures for calculating the tax refund estimation.
12.  *****/
13. #import "ConstantsTax.h"
14.
15. NSInteger    const PersonalSingleTaxBandMax          = 36400;
16. NSInteger    const PersonalSingleTaxCredit           = 3660;
17.
18. NSInteger    const PersonalMarriedTaxBandMax         = 45400;

```

```

19.  NSInteger    const PersonalMarriedTaxCredit      = 5490;
20.  NSInteger    const PersonalSpouseTaxBandMax    = 27400;
21.  NSInteger    const PersonalSpouseTaxCredit     = 1830;
22.
23.  NSInteger    const PersonalSingleParentTaxBandMax  = 40400;
24.  NSInteger    const PersonalSingleParentTaxCredit  = 5490;
25.
26.  NSInteger    const HomeCarerTaxCredit            = 900;
27.  NSInteger    const HomeCarerSpouseAnnualSalaryMax = 5080;
28.
29.  NSInteger    const BinChargesMax                = 400;
30.  NSInteger    const TradeUnionCredit             = 70;
31.  NSInteger    const rentReliefSingleU65Max       = 2000;
32.  NSInteger    const rentReliefSingleO65Max      = 4000;
33.  NSInteger    const rentReliefMarriedU65Max     = 4000;
34.  NSInteger    const rentReliefMarriedO65Max     = 8000;
35.  NSInteger    const taxYear                     = 2009;
36.
37.  float        const TaxRateMax                  = 0.2;
38.  float        const TaxRateMin                  = 0.41;
39.  float        const TaxRateBin                   = 0.2;
40.  float        const TaxRateRentRelief           = 0.2;
41.
42.  @implementation ConstantsTax
43.
44.
45.  @end

```

2.19. CalculateTaxRefund.h

```
1. //
2. // CalculateTaxRefund.h
3. // TaxCalc2009
4. //
5. // Created by C00115609 on 02/03/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8.
9. #import <Foundation/Foundation.h>
10. @class SwitchViewController;
11.
12. @interface CalculateTaxRefund : NSObject {
13.
14.     int incomeAfterDeductionMedical, incomeAfterDeductionDental, taxBands, taxBills, refund, tmp,
        noOfCalcSegments;
15.     int taxBandsSpouse1, taxBandsSpouse2, totalMarriedTaxCredit;
16.     int annualSalary, spouseAnnualSalary, taxOnSalary, spouseTaxOnSalary, allowableMedical,
        allowableDental, binCharges;
17.     int intDateOfBirth, annualRent, totalTaxOnSalary;
18.     NSString *rentFrom, *strDateOfBirth, *rentOrMorgage, *strMaritalStatus;
19.     BOOL tradeUnion, spouseTradeUnion, dependentChild, cohabiting;
20. }
21.
22. - (void) taxRefundCalculation: (SwitchViewController *) switchController;
23. - (void) getUserInformation: (SwitchViewController *) switchController;
24.
25. - (void) taxBandSingle: (SwitchViewController *) switchController;
26. - (void) taxBandSingleParent: (SwitchViewController *) switchController;
27. - (void) taxBandMarried: (SwitchViewController *) switchController;
28. - (void) taxBandMarriedAndDepChild: (SwitchViewController *) switchController;
29. - (void) changeRefundLabelText: (SwitchViewController *) switchController;
30.
31. - (NSString*) commaSeparateThousands: (int) number;
32.
```


33. @end

2.20. CalculateTaxRefund.m

```
1. //
2. // CalculateTaxRefund.m
3. // TaxCalc2009
4. //
5. // Created by C00115609 on 02/03/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8. /*****
9.             Class Description
10.  This class is responsible for calculating the tax refund estimation.
11.  *****/
12.
13. #import "CalculateTaxRefund.h"
14. #import "SwitchViewController.h"
15. #import "ApplyViewController.h"
16. #import "RefundResultViewController.h"
17. #import "SpouseViewController.h"
18. #import "ExpensesViewController.h"
19. #import "MyTaxViewController.h"
20. #import "DetailsViewController.h"
21. #import "DetailsViewController.h"
22. #import "ConstantsTax.h"
23.
24. @implementation CalculateTaxRefund
25.
26. - (void)taxRefundCalculation: (SwitchViewController *) switchController
27. {
28.     [self getUserInformation: switchController];
29.
30.     // Income Deductions
31.     incomeAfterDeductionMedical = annualSalary - allowableMedical;
```

```

32.
33.     if (allowableDental > incomeAfterDeductionMedical) {
34.         if (noOfCalcSegments == 4) {
35.             // Sum of the tax on salary (Add Spouse's tax on salary)
36.             totalTaxOnSalary = [[NSNumber numberWithInt:taxOnSalary] intValue] + [[NSNumber
numberWithInt:spouseTaxOnSalary] intValue];
37.             switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
 stringByAppendingString:[self commaSeparateThousands:totalTaxOnSalary]];
38.             [self changeRefundLabelText:switchController];
39.             return;
40.         }
41.         else { // noOfCalcSegments == 3
42.             switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
 stringByAppendingString:[self commaSeparateThousands:taxOnSalary]];
43.             [self changeRefundLabelText:switchController];
44.             return;
45.         }
46.     }
47.     else {
48.         incomeAfterDeductionDental = incomeAfterDeductionMedical - allowableDental;
49.     }
50.     ////////////////////////////////////////
51.
52.
53.     if ([strMaritalStatus isEqualToString:@"Single"]) {
54.         // Calculate the tax bands (20% up to 36,400 and 41% after FOR SINGLE)
55.         if (switchController.detailsViewController.cohabitingSwitch.hidden == YES
56.             || (dependentChild == YES && cohabiting == YES)){
57.
58.             [self taxBandSingle: switchController];
59.
60.             // If enter else, Calculate the User Tax Refund Estimation
61.             if (taxBands > PersonalSingleTaxCredit) {
62.                 taxBills = taxBands - PersonalSingleTaxCredit;
63.             }
64.             else {

```

```

65.                 switchController.refundResultViewController.refundEstimateTextField.text =
["@€" stringByAppendingString:[self commaSeparateThousands:taxOnSalary]];
66.                 [self changeRefundLabelText:switchController];
67.                 return;
68.             }
69.         }
70.         // Calculate the tax bands (20% up to 40,400 and 41% after FOR SINGLE PARENT-has
Dependent Children)
71.         else if ((dependentChild == YES && cohabiting == NO)){
72.             [self taxBandSingleParent: switchController];
73.
74.             // If enter else, Calculate the User Tax Refund Estimation
75.             if (taxBands > PersonalSingleParentTaxCredit) {
76.                 taxBills = taxBands - PersonalSingleParentTaxCredit;
77.             }
78.             else {
79.                 switchController.refundResultViewController.refundEstimateTextField.text =
["@€" stringByAppendingString:[self commaSeparateThousands:taxOnSalary]];
80.                 [self changeRefundLabelText:switchController];
81.                 return;
82.             }
83.         }
84.     }
85.
86.     else if ([strMaritalStatus isEqualToString:@"Married"]) {
87.
88.         // Calculate the tax bands (20% up to 45,400 and 41% after FOR MARRIED)
89.         if (dependentChild == NO || (dependentChild == YES
90.         && spouseAnnualSalary >= HomeCarerSpouseAnnualSalaryMax)) {
91.             [self taxBandMarried: switchController];
92.
93.             // If enter else, Calculate the User Tax Refund Estimation
94.             if (taxBands > totalMarriedTaxCredit) {
95.                 taxBills = taxBands - totalMarriedTaxCredit;
96.             }
97.             else {

```

```

98.                // Sum of the tax on salary (Add Spouse's tax on salary)
99.                totalTaxOnSalary = [[NSNumber numberWithInt:taxOnSalary] intValue] +
[[NSNumber numberWithInt:spouseTaxOnSalary] intValue];
100.
    switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:totalTaxOnSalary]];
101.                [self changeRefundLabelText:switchController];
102.                return;
103.            }
104.        }
105.        // has dependent children (Additional Home Carer Tax Credit 900)
106.        else if (dependentChild == YES && spouseAnnualSalary <
HomeCarerSpouseAnnualSalaryMax) {
107.            [self taxBandMarriedAndDepChild: switchController];
108.
109.            // If enter else, Calculate the User Tax Refund Estimation
110.            if (taxBands > totalMarriedTaxCredit) {
111.                taxBills = taxBands - totalMarriedTaxCredit;
112.            }
113.            else {
114.                // Sum of the tax on salary (Add Spouse's tax on salary)
115.                totalTaxOnSalary = [[NSNumber numberWithInt:taxOnSalary] intValue] +
[[NSNumber numberWithInt:spouseTaxOnSalary] intValue];
116.
    switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:totalTaxOnSalary]];
117.                [self changeRefundLabelText:switchController];
118.                return;
119.            }
120.        }
121.    }
122.    //////////////////////////////////////
////////
123.
124.
125.    // Tax Bills after the Tax Credits (Bin charges, trade union)
126.    if (binCharges != 0) {

```

```

127.
128.         if (binCharges >= BinChargesMax)
129.             taxBills = taxBills - (BinChargesMax * TaxRateBin);
130.         else
131.             taxBills = taxBills - (binCharges * TaxRateBin);
132.     }
133.     //////////////////////////////////////
134.     ///////////////
135.
136.     // Tax Bills after member of trade union
137.     if (tradeUnion == YES) {
138.         taxBills = taxBills - TradeUnionCredit;
139.     }
140.
141.     if (noOfCalcSegments == 4 && spouseTradeUnion == YES) {
142.         taxBills = taxBills - TradeUnionCredit;
143.     }
144.     //////////////////////////////////////
145.     ///////////////
146.
147.     // The user have rent relief
148.     if ([rentOrMorgage isEqualToString:@"Rent"] && [rentFrom isEqualToString:@"Private landloard"])
149.     {
150.         // Find the user age
151.         strDateOfBirth = [strDateOfBirth substringWithRange:NSMakeRange([strDateOfBirth length] -
152.         2, 2)];
153.         strDateOfBirth = [@"19" stringByAppendingString:strDateOfBirth];
154.         intDateOfBirth = [strDateOfBirth intValue];
155.         intDateOfBirth = taxYear - intDateOfBirth;
156.
157.         // The user is single and age under 65
158.         if ([strMaritalStatus isEqualToString:@"Single"] && intDateOfBirth <= 65) {

```

```

159.         if (annualRent <= rentReliefSingleU65Max) {
160.             if (taxBills >= (annualRent * TaxRateRentRelief)) {
161.                 taxBills = taxBills - (annualRent * TaxRateRentRelief);
162.             }
163.             else {
164.
switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:taxOnSalary]];
165.                 [self changeRefundLabelText:switchController];
166.                 return;
167.             }
168.         }
169.         else {
170.             if (taxBills >= (rentReliefSingleU65Max * TaxRateRentRelief)) {
171.                 taxBills = taxBills - (rentReliefSingleU65Max * TaxRateRentRelief);
172.             }
173.             else {
174.
switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:taxOnSalary]];
175.                 [self changeRefundLabelText:switchController];
176.                 return;
177.             }
178.         }
179.     }
180.     // The user is single and age over 65
181.     else if ([strMaritalStatus isEqualToString:@"Single"] && intDateOfBirth > 65) {
182.
183.         if (annualRent <= rentReliefSingleO65Max) {
184.             if (taxBills >= (annualRent * TaxRateRentRelief)) {
185.                 taxBills = taxBills - (annualRent * TaxRateRentRelief);
186.             }
187.             else {
188.
switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:taxOnSalary]];

```

```

189.             [self changeRefundLabelText:switchController];
190.             return;
191.         }
192.     }
193.     else {
194.         if (taxBills >= (rentReliefSingleO65Max * TaxRateRentRelief)) {
195.             taxBills = taxBills - (rentReliefSingleO65Max * TaxRateRentRelief);
196.         }
197.         else {
198.             switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:taxOnSalary]];
199.             [self changeRefundLabelText:switchController];
200.             return;
201.         }
202.     }
203. }
204. // The user is married and age under 65
205. else if ([strMaritalStatus isEqualToString:@"Married"] && intDateOfBirth <= 65) {
206.
207.     if (annualRent <= rentReliefMarriedU65Max) {
208.         if (taxBills >= (annualRent * TaxRateRentRelief)) {
209.             taxBills = taxBills - (annualRent * TaxRateRentRelief);
210.         }
211.         else {
212.             // Sum of the tax on salary (Add Spouse's tax on salary)
213.             totalTaxOnSalary = [[NSNumber numberWithInt:taxOnSalary] intValue] +
[[NSNumber numberWithInt:spouseTaxOnSalary] intValue];
214.             switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:totalTaxOnSalary]];
215.             [self changeRefundLabelText:switchController];
216.             return;
217.         }
218.     }
219.     else {

```

```

220.         if (taxBills >= (rentReliefMarriedU65Max * TaxRateRentRelief)) {
221.             taxBills = taxBills - (rentReliefMarriedU65Max * TaxRateRentRelief);
222.         }
223.         else {
224.             // Sum of the tax on salary (Add Spouse's tax on salary)
225.             totalTaxOnSalary = [[NSNumber numberWithInt:taxOnSalary] intValue] +
[[NSNumber numberWithInt:spouseTaxOnSalary] intValue];
226.         switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:totalTaxOnSalary]];
227.             [self changeRefundLabelText:switchController];
228.             return;
229.         }
230.     }
231. }
232. // The user is married and age over 65
233. else if ([strMaritalStatus isEqualToString:@"Married"] && intDateOfBirth > 65) {
234.
235.     if (annualRent <= rentReliefMarriedO65Max) {
236.         if (taxBills >= (annualRent * TaxRateRentRelief)) {
237.             taxBills = taxBills - (annualRent * TaxRateRentRelief);
238.         }
239.         else {
240.             // Sum of the tax on salary (Add Spouse's tax on salary)
241.             totalTaxOnSalary = [[NSNumber numberWithInt:taxOnSalary] intValue] +
[[NSNumber numberWithInt:spouseTaxOnSalary] intValue];
242.         switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:totalTaxOnSalary]];
243.             [self changeRefundLabelText:switchController];
244.             return;
245.         }
246.     }
247.     else {
248.         if (taxBills >= (rentReliefMarriedO65Max * TaxRateRentRelief)) {
249.             taxBills = taxBills - (rentReliefMarriedO65Max * TaxRateRentRelief);

```



```

279.             switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:[self commaSeparateThousands:refund]];
280.             [self changeRefundLabelText:switchController];
281.         }
282.         else {
283.             switchController.refundResultViewController.refundEstimateTextField.text = [@"€"
stringByAppendingString:@"0"];
284.             [self changeRefundLabelText:switchController];
285.         }
286.     }
287.     //////////////////////////////////////
288.
289. }
290.
291. // Calculate the tax bands (20% up to 36,400 and 41% after FOR SINGLE)
292. - (void) taxBandSingle: (SwitchViewController *) switchController
293. {
294.     if (incomeAfterDeductionDental <= PersonalSingleTaxBandMax) {
295.         taxBands = incomeAfterDeductionDental * TaxRateMax;
296.     }
297.     else {
298.         tmp = incomeAfterDeductionDental - PersonalSingleTaxBandMax;
299.         taxBands = tmp * TaxRateMin;
300.         taxBands = taxBands + (PersonalSingleTaxBandMax * TaxRateMax);
301.     }
302. }
303.
304. - (void) taxBandSingleParent: (SwitchViewController *) switchController
305. {
306.     if (incomeAfterDeductionDental <= PersonalSingleParentTaxBandMax) {
307.         taxBands = incomeAfterDeductionDental * TaxRateMax;
308.     }
309.     else {
310.         tmp = incomeAfterDeductionDental - PersonalSingleParentTaxBandMax;
311.         taxBands = tmp * TaxRateMin;
312.         taxBands = taxBands + (PersonalSingleParentTaxBandMax * TaxRateMax);

```

```

313.     }
314. }
315.
316. // Calculate the tax bands (20% up to 45,400 and 41% after FOR SPOUSE1)
317. // Calculate the tax bands (20% up to 27,400 and 41% after FOR SPOUSE2)
318. - (void) taxBandMarried: (SwitchViewController *) switchController
319. {
320.     // Spouse 1
321.     if (incomeAfterDeductionDental <= PersonalMarriedTaxBandMax) {
322.         taxBandsSpouse1 = incomeAfterDeductionDental * TaxRateMax;
323.     }
324.     else {
325.         tmp = incomeAfterDeductionDental - PersonalMarriedTaxBandMax;
326.         taxBandsSpouse1 = tmp * TaxRateMin;
327.         taxBandsSpouse1 = taxBandsSpouse1 + (PersonalMarriedTaxBandMax * TaxRateMax);
328.     }
329.
330.     // Spouse 2
331.     if (spouseAnnualSalary <= PersonalSpouseTaxBandMax) {
332.         taxBandsSpouse2 = spouseAnnualSalary * TaxRateMax;
333.     }
334.     else {
335.         tmp = spouseAnnualSalary - PersonalSpouseTaxBandMax;
336.         taxBandsSpouse2 = tmp * TaxRateMin;
337.         taxBandsSpouse2 = taxBandsSpouse2 + (PersonalSpouseTaxBandMax * TaxRateMax);
338.     }
339.
340.     // Get the total tax bands
341.     taxBands = (taxBandsSpouse1 + taxBandsSpouse2);
342.
343.     // Calculate the total tax credit
344.     if ((spouseAnnualSalary * TaxRateMax) < PersonalSpouseTaxCredit) {
345.         totalMarriedTaxCredit = PersonalMarriedTaxCredit + (spouseAnnualSalary * TaxRateMax);
346.     }
347.     else {
348.         totalMarriedTaxCredit = PersonalMarriedTaxCredit + PersonalSpouseTaxCredit;

```

```

349.     }
350. }
351.
352. - (void) taxBandMarriedAndDepChild: (SwitchViewController *) switchController
353. {
354.     // Spouse 1
355.     if (incomeAfterDeductionDental <= PersonalMarriedTaxBandMax) {
356.         taxBandsSpouse1 = incomeAfterDeductionDental * TaxRateMax;
357.     }
358.     else {
359.         tmp = incomeAfterDeductionDental - PersonalMarriedTaxBandMax;
360.         taxBandsSpouse1 = tmp * TaxRateMin;
361.         taxBandsSpouse1 = taxBandsSpouse1 + (PersonalMarriedTaxBandMax * TaxRateMax);
362.     }
363.
364.     // Get the total tax bands
365.     taxBands = taxBandsSpouse1;
366.     totalMarriedTaxCredit = PersonalMarriedTaxCredit + HomeCarerTaxCredit;
367. }
368.
369. // Thousand Separator
370. -(NSString*) commaSeparateThousands: (int) number
371. {
372.     NSString *result = @"";
373.
374.     if ( number == 0 ) {
375.         return @"0";
376.     }
377.     else {
378.         while( number > 0 )
379.         {
380.             int removedDigits = number % 1000;
381.             if( result != nil && [result length] > 0 )
382.             {
383.                 if( number > 999 )
384.                     result = [NSString stringWithFormat:@"%%.3i,%@", removedDigits, result];

```

```

385.             else
386.                 result = [NSString stringWithFormat:@"%i,%@", removedDigits, result];
387.         }
388.     else
389.     {
390.         if( number > 999 )
391.             result = [NSString stringWithFormat:@"%i.%3i", removedDigits];
392.         else
393.             result = [NSString stringWithFormat:@"%i", removedDigits];
394.     }
395.     number = number/1000;
396. }
397.
398.     return result;
399. }
400. }
401.
402. // Change the labels of the RefundResult view with appropriate messages
403. - (void) changeRefundLabelText: (SwitchViewController *) switchController;
404. {
405.     if ([[switchController refundResultViewController] refundEstimateTextField] text]
406.         isEqualToString:@"€0"]) {
407.         [[switchController refundResultViewController] greenLabel] setText:[NSString
408.             stringWithString:@"          Sorry!"];
409.         [[switchController refundResultViewController] warningLabel] setText:[NSString
410.             stringWithString:@"  Are you sure you put the correct details in?"];
411.     }
412.     else {
413.         [[switchController refundResultViewController] greenLabel] setText:[NSString
414.             stringWithString:@"Congratulations!"];
415.         [[switchController refundResultViewController] warningLabel] setText:[NSString
416.             stringWithString:@"That's just for 2009 so you could be due more."];
417.     }
418. }
419. }
420. }

```

```

416. // Get the entire user information inputs
417. - (void) getUserInformation: (SwitchViewController *) switchController
418. {
419.     annualSalary = [[[[[switchController myTaxViewController] annualSalaryTextField] text]
    stringByReplacingOccurrencesOfString:@"," withString:@""] intValue];
420.     taxOnSalary = [[[[[switchController myTaxViewController] taxOnPayTextField] text]
    stringByReplacingOccurrencesOfString:@"," withString:@""] intValue];
421.
422.     spouseAnnualSalary = [[[[[switchController spouseViewController] spouseAnnualSalaryTextField]
    text] stringByReplacingOccurrencesOfString:@"," withString:@""] intValue];
423.     spouseTaxOnSalary = [[[[[switchController spouseViewController] spouseTaxOnPayTextField] text]
    stringByReplacingOccurrencesOfString:@"," withString:@""] intValue];
424.
425.     allowableMedical = [[[[[switchController expensesViewController] allowableMedicalTextField]
    text] stringByReplacingOccurrencesOfString:@"," withString:@""] intValue];
426.     allowableDental = [[[[[switchController expensesViewController] allowableDentalTextField] text]
    stringByReplacingOccurrencesOfString:@"," withString:@""] intValue];
427.     binCharges = [[[[[switchController expensesViewController] binChargesTextField] text]
    stringByReplacingOccurrencesOfString:@"," withString:@""] intValue];
428.     tradeUnion = [[switchController myTaxViewController] tradeUnionSwitch] isOn];
429.     spouseTradeUnion = [[switchController spouseViewController] spouseTradeUnionSwitch] isOn];
430.
431.     rentFrom = [[[[[switchController expensesViewController] rentFromPickerButton] titleLabel]
    text] stringByTrimmingCharactersInSet:[NSCharacterSet whitespaceAndNewlineCharacterSet]];
432.     strMaritalStatus = [[[[[switchController detailsViewController] maritalStatusPickerButton]
    titleLabel] text] stringByTrimmingCharactersInSet:[NSCharacterSet whitespaceAndNewlineCharacterSet]];
433.     strDateOfBirth = [[[[[switchController detailsViewController] dateOfBirthPickerButton]
    titleLabel] text] stringByTrimmingCharactersInSet:[NSCharacterSet whitespaceAndNewlineCharacterSet]];
434.     rentOrMorgage = [[[[[switchController expensesViewController] rentOrMorgagePickerButton]
    titleLabel] text] stringByTrimmingCharactersInSet:[NSCharacterSet whitespaceAndNewlineCharacterSet]];
435.     annualRent = [[[[[switchController expensesViewController] annualRentTextField] text]
    stringByReplacingOccurrencesOfString:@"," withString:@""] intValue];
436.     dependentChild = [[switchController detailsViewController] dependentChildrenSwitch] isOn];
437.     cohabiting = [[switchController detailsViewController] cohabitingSwitch] isOn];
438.
439.     noOfCalcSegments = [[switchController calculatorSegment] numberOfSegments];
440. }

```

```
441.  
442. @end
```

2.21. ResetCalculator.h

```
1. //  
2. // ResetCalculator.h  
3. // TaxCalc2009  
4. //  
5. // Created by C00115609 on 29/03/2010.  
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.  
7. //  
8.  
9. #import <Foundation/Foundation.h>  
10. @class SwitchViewController;  
11.  
12. @interface ResetCalculator : NSObject {  
13.  
14. }  
15.  
16. - (void)resetCalculatorInfo: (SwitchViewController *) switchController;  
17. @end
```

2.22. ResetCalculator.m

```
1. //
2. // ResetCalculator.m
3. // TaxCalc2009
4. //
5. // Created by C00115609 on 29/03/2010.
6. // Copyright 2010 __Tuna Erdurmaz__. All rights reserved.
7. //
8. /*****
9.                                     Class Description
10.  This class is responsible for bringing an application to initial state.
11.  *****/
12.
13. #import "ResetCalculator.h"
14. #import "CalculateTaxRefund.h"
15. #import "SwitchViewController.h"
16. #import "ApplyViewController.h"
17. #import "RefundResultViewController.h"
18. #import "SpouseViewController.h"
19. #import "ExpensesViewController.h"
20. #import "MyTaxViewController.h"
21. #import "DetailsViewController.h"
22. #import "DetailsViewController.h"
23.
24. @implementation ResetCalculator
25.
26. - (void)resetCalculatorInfo: (SwitchViewController *) switchController
27. {
28.     // Clear the DetailsView
29.     [[[switchController detailsViewController] maritalStatusPickerButton]
30.     setTitle:@"Single" forState:UIControlStateNormal];
31.     [[[switchController detailsViewController] pickerViewDetails]
32.     selectRow:0 inComponent:0 animated:NO];
33.     [[[switchController detailsViewController] dateOfBirthPickerButton]
34.     setTitle:@" Select.." forState:UIControlStateNormal];
```



```

35.     [switchController maritalStatus:[switchController detailsViewController]];
36.     switchController.detailsViewController.strMaritalStatus =
37.     [[NSString alloc] initWithString:@"Single"];
38.     [[[switchController detailsViewController] dependentChildrenSwitch] setOn:NO];
39.     [[[switchController detailsViewController] cohabitingSwitch] setOn:NO];
40.     [[[switchController detailsViewController] presentCohabitingInfo];
41.
42.
43.     // Clear the MyTaxView
44.     [[[switchController myTaxViewController] annualSalaryTextField] setText:nil];
45.     [[[switchController myTaxViewController] taxOnPayTextField] setText:nil];
46.     [[[switchController myTaxViewController] tradeUnionSwitch] setOn:NO];
47.
48.     // Clear the ExpensesView
49.     [[[switchController expensesViewController] allowableMedicalTextField] setText:nil];
50.     [[[switchController expensesViewController] allowableDentalTextField] setText:nil];
51.     [[[switchController expensesViewController] binChargesTextField] setText:nil];
52.     [[[switchController expensesViewController] rentOrMorgagePickerButton]
53.     setTitle:@"Neither " forState:UIControlStateNormal];
54.     [[[switchController expensesViewController] pickerViewRentOrMortgage]
55.     selectRow:0 inComponent:0 animated:NO];
56.     [[[switchController expensesViewController] pickerViewRentFrom]
57.     selectRow:0 inComponent:0 animated:NO];
58.     [[[switchController expensesViewController] interestReliefSwitch] setOn:NO];
59.     [[[switchController expensesViewController] rentFromPickerButton]
60.     setTitle:@"      Select..          " forState:UIControlStateNormal];
61.     switchController.expensesViewController.rentFromLabel.hidden = YES;
62.     switchController.expensesViewController.blankImageView.hidden = YES;
63.     switchController.expensesViewController.rentFromPickerButton.hidden = YES;
64.     switchController.expensesViewController.pickerDownArrowImageView.hidden = YES;
65.     switchController.expensesViewController.rentFromHelpButton.hidden = YES;
66.     [[[switchController expensesViewController] annualRentTextField] setText:nil];
67.     switchController.expensesViewController.annualRentLabel.hidden = YES;
68.     switchController.expensesViewController.annualRentEuroLabel.hidden = YES;
69.     switchController.expensesViewController.annualRentTextField.hidden = YES;
70.     switchController.expensesViewController.annualRentHelpButton.hidden = YES;

```

```
71.     switchController.expensesViewController.interestReliefLabel.hidden = YES;
72.     switchController.expensesViewController.interestReliefSwitch.hidden = YES;
73.     switchController.expensesViewController.interestReliefHelpButton.hidden = YES;
74.     switchController.expensesViewController.redOakManImageView.hidden = NO;
75.
76.     // Clear the SpouseView
77.     [[[switchController spouseViewController] dateOfBirthPickerButton]
78.     setTitle:@" Select..          " forState:UIControlStateNormal];
79.     [[[switchController spouseViewController] spouseAnnualSalaryTextField] setText:nil];
80.     [[[switchController spouseViewController] spouseTaxOnPayTextField] setText:nil];
81.     [[[switchController spouseViewController] spouseTradeUnionSwitch] setOn:NO];
82. }
83.
84. @end
```