```
 1:
 2: // Vahe Karamian - Software Engineering - Copyright 2001
 3: // Hotel Information Management System
 4:
 5: //---------------------------------------------------------------------------
 6:
 7: #ifndef frmEmployeeScheduleH
 8: #define frmEmployeeScheduleH
 9: //---------------------------------------------------------------------------
10: #include <Classes.hpp>
11: #include <Controls.hpp>
12: #include <StdCtrls.hpp>
13: #include <Forms.hpp>
14: #include <Buttons.hpp>
15: #include <DBCtrls.hpp>
16: #include <DBGrids.hpp>
17: #include <ExtCtrls.hpp>
18: #include <Grids.hpp>
19: #include <Mask.hpp>
20: #include <ComCtrls.hpp>
21: //---------------------------------------------------------------------------
22: class TemployeeSchedule : public TForm
23: {
24: __published:   // IDE-managed Components
25:    TBitBtn *butClose;
26:    TLabel *Label1;
27:    TLabel *Label2;
28:    TLabel *Label3;
29:    TLabel *Label4;
30:    TDBEdit *dbtxtSSN;
31:    TDBEdit *dbtxtStartTime;
32:    TDBEdit *dbtxtEndTime;
33:    TDBComboBox *DBComboBox1;
34:    TDBGrid *dbGridEmployeeSchedule;
35:    TDBNavigator *dbEmployeeScheduleNavigator;
36:    void __fastcall butCloseClick(TObject *Sender);
37: private: // User declarations
38: public:     // User declarations
39:    __fastcall TemployeeSchedule(TComponent* Owner);
40: };
41: //---------------------------------------------------------------------------
42: extern PACKAGE TemployeeSchedule *employeeSchedule;
43: //---------------------------------------------------------------------------
44: #endif
45:
46:
```

```
47: //---------------------------------------------------------------------------
48:
49: #ifndef frmEmployeeInformationH
50: #define frmEmployeeInformationH
51: //---------------------------------------------------------------------------
52: #include <Classes.hpp>
53: #include <Controls.hpp>
54: #include <StdCtrls.hpp>
55: #include <Forms.hpp>
56: #include <Buttons.hpp>
57: #include <DBCtrls.hpp>
58: #include <Mask.hpp>
59: //---------------------------------------------------------------------------
60: class TemployeeInformation : public TForm
61: {
62: __published:   // IDE-managed Components
63:    TLabel *Label1;
64:    TLabel *Label2;
65:    TLabel *Label3;
66:    TLabel *Label4;
67:    TLabel *Label5;
68:    TLabel *Label6;
69:    TLabel *Label8;
70:    TLabel *Label9;
71:    TLabel *Label10;
72:    TBitBtn *butSubmit;
73:    TBitBtn *butClose;
74:    TBitBtn *butDelete;
75:    TEdit *txtSSN;
76:    TEdit *txtFirstName;
77:    TEdit *txtLastName;
78:    TEdit *txtAddress;
79:    TEdit *txtCity;
80:    TEdit *txtState;
81:    TEdit *txtPhone;
82:    TEdit *txtTitle;
83:    TEdit *txtRate;
84:    TCheckBox *cbSalary;
85:    TBitBtn *butNewCustomer;
86:    TBitBtn *butSSN;
87:    void __fastcall butCloseClick(TObject *Sender);
88:    void __fastcall butNewCustomerClick(TObject *Sender);
89:    void __fastcall butSSNClick(TObject *Sender);
90:    void __fastcall butSubmitClick(TObject *Sender);
91:    void __fastcall butDeleteClick(TObject *Sender);
92:    void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
```

```
 93: private: // User declarations
 94: public:     // User declarations
 95:    __fastcall TemployeeInformation(TComponent* Owner);
 96: };
 97: //---------------------------------------------------------------------------
 98: extern PACKAGE TemployeeInformation *employeeInformation;
 99: //---------------------------------------------------------------------------
100: #endif
101:
102:
103: //---------------------------------------------------------------------------
104:
105: #ifndef frmCustomerH
106: #define frmCustomerH
107: //---------------------------------------------------------------------------
108: #include <Classes.hpp>
109: #include <Controls.hpp>
110: #include <StdCtrls.hpp>
111: #include <Forms.hpp>
112: #include <Buttons.hpp>
113: //---------------------------------------------------------------------------
114: class TcustomerWindow : public TForm
115: {
116: __published:   // IDE-managed Components
117:    TLabel *Label1;
118:    TEdit *txtCustomerID;
119:    TLabel *Label2;
120:    TEdit *txtAccountID;
121:    TBitBtn *butCustomerID;
122:    TLabel *Label3;
123:    TEdit *txtFirstName;
124:    TLabel *Label4;
125:    TEdit *txtLastName;
126:    TLabel *Label5;
127:    TEdit *txtAddress;
128:    TLabel *Label6;
129:    TEdit *txtCity;
130:    TLabel *Label7;
131:    TEdit *txtState;
132:    TLabel *Label8;
133:    TEdit *txtZipCode;
134:    TLabel *Label9;
135:    TEdit *txtPhone;
136:    TBitBtn *butSubmit;
137:    TBitBtn *butClose;
138:    TBitBtn *butAccountID;
```

```
139:    TBitBtn *butNewCustomer;
140:    void __fastcall butCloseClick(TObject *Sender);
141:    void __fastcall butAccountIDClick(TObject *Sender);
142:    void __fastcall butCustomerIDClick(TObject *Sender);
143:    void __fastcall butNewCustomerClick(TObject *Sender);
144:    void __fastcall butSubmitClick(TObject *Sender);
145: private: // User declarations
146: public:     // User declarations
147:    __fastcall TcustomerWindow(TComponent* Owner);
148: };
149: //---------------------------------------------------------------------------
150: extern PACKAGE TcustomerWindow *customerWindow;
151: //---------------------------------------------------------------------------
152: #endif
153:
154:
155: //---------------------------------------------------------------------------
156:
157: #ifndef frmCalendarH
158: #define frmCalendarH
159: //---------------------------------------------------------------------------
160: #include <Classes.hpp>
161: #include <Controls.hpp>
162: #include <StdCtrls.hpp>
163: #include <Forms.hpp>
164: #include "CCALENDR.h"
165: #include <Grids.hpp>
166: #include <Buttons.hpp>
167: //---------------------------------------------------------------------------
168: class TcalendarWindow : public TForm
169: {
170: __published:   // IDE-managed Components
171:    TCCalendar *globalCalendar;
172:    TBitBtn *BitBtn1;
173:    void __fastcall BitBtn1Click(TObject *Sender);
174: private: // User declarations
175: public:     // User declarations
176:    __fastcall TcalendarWindow(TComponent* Owner);
177: };
178: //---------------------------------------------------------------------------
179: extern PACKAGE TcalendarWindow *calendarWindow;
180: //---------------------------------------------------------------------------
181: #endif
182:
183:
184: //---------------------------------------------------------------------------
```

```
185:
186: #ifndef frmAccountH
187: #define frmAccountH
188: //---------------------------------------------------------------------------
189: #include <Classes.hpp>
190: #include <Controls.hpp>
191: #include <StdCtrls.hpp>
192: #include <Forms.hpp>
193: #include <Buttons.hpp>
194: #include <ComCtrls.hpp>
195: //---------------------------------------------------------------------------
196: class TaccountWindow : public TForm
197: {
198: __published:   // IDE-managed Components
199:    TLabel *Label1;
200:    TEdit *txtAccountID;
201:    TLabel *Label2;
202:    TEdit *txtCompanyName;
203:    TLabel *Label3;
204:    TEdit *txtCreditCardNumber;
205:    TLabel *Label5;
206:    TEdit *txtAddress;
207:    TLabel *Label6;
208:    TEdit *txtCity;
209:    TLabel *Label7;
210:    TEdit *txtState;
211:    TLabel *Label8;
212:    TEdit *txtZipCode;
213:    TBitBtn *butSubmit;
214:    TBitBtn *BitBtn2;
215:    TLabel *Label4;
216:    TBitBtn *butNewAccount;
217:    TLabel *Label9;
218:    TEdit *txtBalance;
219:    TBitBtn *butAccountNumber;
220:    TEdit *txtExpirationDate;
221:    void __fastcall BitBtn2Click(TObject *Sender);
222:    void __fastcall butNewAccountClick(TObject *Sender);
223:    void __fastcall butAccountNumberClick(TObject *Sender);
224:    void __fastcall butSubmitClick(TObject *Sender);
225: private: // User declarations
226: public:     // User declarations
227:    __fastcall TaccountWindow(TComponent* Owner);
228: };
229: //---------------------------------------------------------------------------
230: extern PACKAGE TaccountWindow *accountWindow;
```

```
231: //---------------------------------------------------------------------------
232: #endif
233:
234:
235: //---------------------------------------------------------------------------
236:
237: #ifndef dataModuleH
238: #define dataModuleH
239: //---------------------------------------------------------------------------
240: #include <Classes.hpp>
241: #include <Controls.hpp>
242: #include <StdCtrls.hpp>
243: #include <Forms.hpp>
244: #include <Db.hpp>
245: #include <DBTables.hpp>
246: //---------------------------------------------------------------------------
247: class TdmHMS : public TDataModule
248: {
249:    __published:   // IDE-managed Components
250:       TDataSource *loginDataSource;
251:       TTable *loginTable;
252:    TDataSource *timeCardDataSource;
253:    TTable *timeCardTable;
254:    TDataSource *reservationDataSource;
255:    TTable *reservationTable;
256:    TDataSource *customerDataSource;
257:    TTable *customerTable;
258:    TDataSource *accountDataSource;
259:    TTable *accountTable;
260:    TDataSource *employeeDataSource;
261:    TTable *employeeTable;
262:    TDataSource *roomInfoDataSource;
263:    TTable *roomInfoTable;
264:    TDataSource *employeeScheduleDataSource;
265:    TTable *employeeScheduleTable;
266:    TDataSource *transactionDataSource;
267:    TTable *transactionTable;
268:
269:    private:       // User declarations
270:
271:    public:        // User declarations
272:       __fastcall TdmHMS(TComponent* Owner);
273:
274:       bool __fastcall LGcheckUser( AnsiString u, AnsiString p );
275:       bool __fastcall RWcheckReservation( int id );
276:       bool __fastcall CWcheckCustomer( int id );
```

```
277:       bool __fastcall AWcheckAccount( int id );
278:       bool __fastcall EWcheckEmployee( AnsiString ssn );
279:
280:       int  __fastcall LGtypeOfUser( AnsiString u );
281:
282:       AnsiString __fastcall LGuserSSN( AnsiString u );
283:       AnsiString __fastcall LGstartTime( AnsiString u );
284:
285:       void __fastcall LGstampTimeCard( AnsiString ssn );
286:       void __fastcall MWstampTimeCard( AnsiString ssn, AnsiString stime );
287:
288:       void __fastcall RWsaveReservation( AnsiString customer,
289:                                          AnsiString employee,
290:                                          AnsiString rate,
291:                                          AnsiString room,
292:                                          AnsiString start,
293:                                          AnsiString end );
294:
295:       void __fastcall CWsaveCustomer( AnsiString accountID,
296:                                       AnsiString fistName,
297:                                       AnsiString lastName,
298:                                       AnsiString address,
299:                                       AnsiString city,
300:                                       AnsiString state,
301:                                       AnsiString zip,
302:                                       AnsiString phone );
303:
304:       void __fastcall AWsaveAccount( AnsiString company,
305:                                      AnsiString creaditCard,
306:                                      AnsiString expiration,
307:                                      AnsiString address,
308:                                      AnsiString city,
309:                                      AnsiString state,
310:                                      AnsiString zip );
311:
312:      void __fastcall EWsaveEmployee(  AnsiString ssn,
313:                                       AnsiString firstName,
314:                                       AnsiString lastName,
315:                                       AnsiString address,
316:                                       AnsiString city,
317:                                       AnsiString state,
318:                                       AnsiString phone,
319:                                       AnsiString title,
320:                                       bool salary,
321:                                       AnsiString rate );
322:
```

```
323:      void __fastcall EWdeleteEmployee( AnsiString ssn );
324:
325:      void __fastcall TWsaveTransaction(   AnsiString customer,
326:                                           AnsiString service,
327:                                           AnsiString payment,
328:                                           AnsiString amount );
329:
330: };
331: //---------------------------------------------------------------------------
332: extern PACKAGE TdmHMS *dmHMS;
333: //---------------------------------------------------------------------------
334: #endif
335:
336: //---------------------------------------------------------------------------
337:
338: #ifndef frmFoodServiceH
339: #define frmFoodServiceH
340: //---------------------------------------------------------------------------
341: #include <Classes.hpp>
342: #include <Controls.hpp>
343: #include <StdCtrls.hpp>
344: #include <Forms.hpp>
345: #include <DBCtrls.hpp>
346: #include <DBGrids.hpp>
347: #include <Grids.hpp>
348: #include <Mask.hpp>
349: #include <Buttons.hpp>
350: //---------------------------------------------------------------------------
351: class TfoodService : public TForm
352: {
353: __published:   // IDE-managed Components
354:    TLabel *Label1;
355:    TLabel *Label2;
356:    TLabel *Label3;
357:    TDBEdit *DBEdit1;
358:    TDBEdit *DBEdit2;
359:    TDBEdit *DBEdit3;
360:    TDBGrid *DBGrid1;
361:    TBitBtn *BitBtn1;
362:    void __fastcall BitBtn1Click(TObject *Sender);
363: private: // User declarations
364: public:     // User declarations
365:    __fastcall TfoodService(TComponent* Owner);
366: };
367: //---------------------------------------------------------------------------
368: extern PACKAGE TfoodService *foodService;
```

```
369: //---------------------------------------------------------------------------
370: #endif
371:
372: //---------------------------------------------------------------------------
373:
374: #ifndef frmEmployeeScheduleH
375: #define frmEmployeeScheduleH
376: //---------------------------------------------------------------------------
377: #include <Classes.hpp>
378: #include <Controls.hpp>
379: #include <StdCtrls.hpp>
380: #include <Forms.hpp>
381: #include <Buttons.hpp>
382: #include <DBCtrls.hpp>
383: #include <DBGrids.hpp>
384: #include <ExtCtrls.hpp>
385: #include <Grids.hpp>
386: #include <Mask.hpp>
387: #include <ComCtrls.hpp>
388: //---------------------------------------------------------------------------
389: class TemployeeSchedule : public TForm
390: {
391: __published:   // IDE-managed Components
392:    TBitBtn *butClose;
393:    TLabel *Label1;
394:    TLabel *Label2;
395:    TLabel *Label3;
396:    TLabel *Label4;
397:    TDBEdit *dbtxtSSN;
398:    TDBEdit *dbtxtStartTime;
399:    TDBEdit *dbtxtEndTime;
400:    TDBComboBox *DBComboBox1;
401:    TDBGrid *dbGridEmployeeSchedule;
402:    TDBNavigator *dbEmployeeScheduleNavigator;
403:    void __fastcall butCloseClick(TObject *Sender);
404: private: // User declarations
405: public:     // User declarations
406:    __fastcall TemployeeSchedule(TComponent* Owner);
407: };
408: //---------------------------------------------------------------------------
409: extern PACKAGE TemployeeSchedule *employeeSchedule;
410: //---------------------------------------------------------------------------
411: #endif
412:
413:
414: //---------------------------------------------------------------------------
```

```
415:
416: #ifndef frmEmployeeInformationH
417: #define frmEmployeeInformationH
418: //---------------------------------------------------------------------------
419: #include <Classes.hpp>
420: #include <Controls.hpp>
421: #include <StdCtrls.hpp>
422: #include <Forms.hpp>
423: #include <Buttons.hpp>
424: #include <DBCtrls.hpp>
425: #include <Mask.hpp>
426: //---------------------------------------------------------------------------
427: class TemployeeInformation : public TForm
428: {
429: __published:   // IDE-managed Components
430:    TLabel *Label1;
431:    TLabel *Label2;
432:    TLabel *Label3;
433:    TLabel *Label4;
434:    TLabel *Label5;
435:    TLabel *Label6;
436:    TLabel *Label8;
437:    TLabel *Label9;
438:    TLabel *Label10;
439:    TBitBtn *butSubmit;
440:    TBitBtn *butClose;
441:    TBitBtn *butDelete;
442:    TEdit *txtSSN;
443:    TEdit *txtFirstName;
444:    TEdit *txtLastName;
445:    TEdit *txtAddress;
446:    TEdit *txtCity;
447:    TEdit *txtState;
448:    TEdit *txtPhone;
449:    TEdit *txtTitle;
450:    TEdit *txtRate;
451:    TCheckBox *cbSalary;
452:    TBitBtn *butNewCustomer;
453:    TBitBtn *butSSN;
454:    void __fastcall butCloseClick(TObject *Sender);
455:    void __fastcall butNewCustomerClick(TObject *Sender);
456:    void __fastcall butSSNClick(TObject *Sender);
457:    void __fastcall butSubmitClick(TObject *Sender);
458:    void __fastcall butDeleteClick(TObject *Sender);
459:    void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
460: private: // User declarations
```

```
461: public:     // User declarations
462:    __fastcall TemployeeInformation(TComponent* Owner);
463: };
464: //---------------------------------------------------------------------------
465: extern PACKAGE TemployeeInformation *employeeInformation;
466: //---------------------------------------------------------------------------
467: #endif
468:
469:
470: //---------------------------------------------------------------------------
471:
472: #ifndef frmCustomerH
473: #define frmCustomerH
474: //---------------------------------------------------------------------------
475: #include <Classes.hpp>
476: #include <Controls.hpp>
477: #include <StdCtrls.hpp>
478: #include <Forms.hpp>
479: #include <Buttons.hpp>
480: //---------------------------------------------------------------------------
481: class TcustomerWindow : public TForm
482: {
483: __published:   // IDE-managed Components
484:    TLabel *Label1;
485:    TEdit *txtCustomerID;
486:    TLabel *Label2;
487:    TEdit *txtAccountID;
488:    TBitBtn *butCustomerID;
489:    TLabel *Label3;
490:    TEdit *txtFirstName;
491:    TLabel *Label4;
492:    TEdit *txtLastName;
493:    TLabel *Label5;
494:    TEdit *txtAddress;
495:    TLabel *Label6;
496:    TEdit *txtCity;
497:    TLabel *Label7;
498:    TEdit *txtState;
499:    TLabel *Label8;
500:    TEdit *txtZipCode;
501:    TLabel *Label9;
502:    TEdit *txtPhone;
503:    TBitBtn *butSubmit;
504:    TBitBtn *butClose;
505:    TBitBtn *butAccountID;
506:    TBitBtn *butNewCustomer;
```

```
507:    void __fastcall butCloseClick(TObject *Sender);
508:    void __fastcall butAccountIDClick(TObject *Sender);
509:    void __fastcall butCustomerIDClick(TObject *Sender);
510:    void __fastcall butNewCustomerClick(TObject *Sender);
511:    void __fastcall butSubmitClick(TObject *Sender);
512: private: // User declarations
513: public:     // User declarations
514:    __fastcall TcustomerWindow(TComponent* Owner);
515: };
516: //---------------------------------------------------------------------------
517: extern PACKAGE TcustomerWindow *customerWindow;
518: //---------------------------------------------------------------------------
519: #endif
520:
521:
522: //---------------------------------------------------------------------------
523:
524: #ifndef transactoinWindowH
525: #define transactoinWindowH
526: //---------------------------------------------------------------------------
527: #include <Classes.hpp>
528: #include <Controls.hpp>
529: #include <StdCtrls.hpp>
530: #include <Forms.hpp>
531: #include <Buttons.hpp>
532: #include <ComCtrls.hpp>
533: //---------------------------------------------------------------------------
534: class TtransactionWindow : public TForm
535: {
536: __published:   // IDE-managed Components
537:    TLabel *Label1;
538:    TLabel *Label2;
539:    TLabel *Label3;
540:    TLabel *Label4;
541:    TLabel *Label5;
542:    TEdit *txtCustomerID;
543:    TEdit *txtServiceID;
544:    TComboBox *cbPaymentMethod;
545:    TEdit *txtAmountDue;
546:    TDateTimePicker *dtDate;
547:    TBitBtn *butSubmit;
548:    TBitBtn *butClose;
549:    TLabel *Label6;
550:    TEdit *txtTime;
551:    void __fastcall butCloseClick(TObject *Sender);
552:    void __fastcall butSubmitClick(TObject *Sender);
```

```
553: private: // User declarations
554: public:     // User declarations
555:     __fastcall TtransactionWindow(TComponent* Owner);
556: };
557: //---------------------------------------------------------------------------
558: extern PACKAGE TtransactionWindow *transactionWindow;
559: //---------------------------------------------------------------------------
560: #endif
561:
562:
563: //---------------------------------------------------------------------------
564:
565: #ifndef frmTimeCardH
566: #define frmTimeCardH
567: //---------------------------------------------------------------------------
568: #include <Classes.hpp>
569: #include <Controls.hpp>
570: #include <StdCtrls.hpp>
571: #include <Forms.hpp>
572: #include <Buttons.hpp>
573: #include <ComCtrls.hpp>
574: #include <DBCtrls.hpp>
575: #include <DBGrids.hpp>
576: #include <ExtCtrls.hpp>
577: #include <Grids.hpp>
578: #include <Mask.hpp>
579: //---------------------------------------------------------------------------
580: class TtimeCard : public TForm
581: {
582: __published:   // IDE-managed Components
583:     TBitBtn *butClose;
584:     TLabel *Label1;
585:     TLabel *Label2;
586:     TLabel *Label3;
587:     TLabel *Label4;
588:     TDBNavigator *dbTimeCardNavigator;
589:     TDBEdit *dbtxtSSN;
590:     TDBEdit *dbtxtStartTime;
591:     TDBEdit *dbtxtDate;
592:     TDBEdit *dbtxtEndTime;
593:     TDBGrid *dbTimeCardGrid;
594:     void __fastcall butCloseClick(TObject *Sender);
595: private: // User declarations
596: public:     // User declarations
597:     __fastcall TtimeCard(TComponent* Owner);
598: };
```

```
599: //---------------------------------------------------------------------------
600: extern PACKAGE TtimeCard *timeCard;
601: //---------------------------------------------------------------------------
602: #endif
603:
604:
605: //---------------------------------------------------------------------------
606:
607: #ifndef frmRoomTypeH
608: #define frmRoomTypeH
609: //---------------------------------------------------------------------------
610: #include <Classes.hpp>
611: #include <Controls.hpp>
612: #include <StdCtrls.hpp>
613: #include <Forms.hpp>
614: #include <Buttons.hpp>
615: #include <DBCtrls.hpp>
616: #include <DBGrids.hpp>
617: #include <ExtCtrls.hpp>
618: #include <Grids.hpp>
619: #include <Mask.hpp>
620: //---------------------------------------------------------------------------
621: class TroomTypeWindow : public TForm
622: {
623: __published:   // IDE-managed Components
624:    TDBGrid *DBGrid1;
625:    TDBNavigator *DBNavigator1;
626:    TDBEdit *dbtxtRoomType;
627:    TDBEdit *DBEdit2;
628:    TDBEdit *DBEdit3;
629:    TDBEdit *DBEdit4;
630:    TLabel *Label1;
631:    TLabel *Label2;
632:    TLabel *Label3;
633:    TLabel *Label4;
634:    TBitBtn *butSubmit;
635:    TBitBtn *butClose;
636:    void __fastcall butCloseClick(TObject *Sender);
637: private: // User declarations
638: public:     // User declarations
639:    __fastcall TroomTypeWindow(TComponent* Owner);
640: };
641: //---------------------------------------------------------------------------
642: extern PACKAGE TroomTypeWindow *roomTypeWindow;
643: //---------------------------------------------------------------------------
644: #endif
```

```
645:
646:
647: //---------------------------------------------------------------------------
648:
649: #ifndef frmRoomInformationH
650: #define frmRoomInformationH
651: //---------------------------------------------------------------------------
652: #include <Classes.hpp>
653: #include <Controls.hpp>
654: #include <StdCtrls.hpp>
655: #include <Forms.hpp>
656: #include <DBCtrls.hpp>
657: #include <DBGrids.hpp>
658: #include <ExtCtrls.hpp>
659: #include <Grids.hpp>
660: #include <Mask.hpp>
661: #include <Buttons.hpp>
662: //---------------------------------------------------------------------------
663: class TroomInformation : public TForm
664: {
665: __published:   // IDE-managed Components
666:    TLabel *Label1;
667:    TLabel *Label2;
668:    TDBCheckBox *dbcbOccupied;
669:    TDBCheckBox *dbcbClean;
670:    TDBEdit *dbtxtRoomNumber;
671:    TDBEdit *dbtxtRoomType;
672:    TDBGrid *dbGridRoomInformation;
673:    TDBNavigator *dbRoomInfoNavigator;
674:    TBitBtn *BitBtn1;
675:    void __fastcall BitBtn1Click(TObject *Sender);
676: private: // User declarations
677: public:     // User declarations
678:    __fastcall TroomInformation(TComponent* Owner);
679: };
680: //---------------------------------------------------------------------------
681: extern PACKAGE TroomInformation *roomInformation;
682: //---------------------------------------------------------------------------
683: #endif
684:
685:
686: //---------------------------------------------------------------------------
687:
688: #ifndef frmReservationH
689: #define frmReservationH
690: //---------------------------------------------------------------------------
```

```
691: #include <Classes.hpp>
692: #include <Controls.hpp>
693: #include <StdCtrls.hpp>
694: #include <Forms.hpp>
695: #include <Buttons.hpp>
696: #include <ComCtrls.hpp>
697: //---------------------------------------------------------------------------
698: class TreservationWindow : public TForm
699: {
700:    __published:   // IDE-managed Components
701:       TLabel *Label1;
702:       TEdit *txtReservationID;
703:       TLabel *Label2;
704:       TLabel *Label4;
705:       TLabel *Label5;
706:       TLabel *Label6;
707:       TLabel *Label7;
708:       TBitBtn *butCustomerInformation;
709:       TBitBtn *butRateRequest;
710:       TBitBtn *butStartDate;
711:       TBitBtn *butEndDate;
712:       TBitBtn *butSubmit;
713:       TBitBtn *butClose;
714:       TBitBtn *butDelete;
715:    TDateTimePicker *dtStartDate;
716:    TDateTimePicker *dtEndDate;
717:
718:       TEdit *txtCustomerID;
719:    TEdit *txtRateID;
720:       TEdit *txtRoomNumber;
721:
722:       TBitBtn *butReservation;
723:       TBitBtn *butNewReservation;
724:
725:       void __fastcall butCustomerInformationClick(TObject *Sender);
726:       void __fastcall butCloseClick(TObject *Sender);
727:       void __fastcall butRateRequestClick(TObject *Sender);
728:    void __fastcall butReservationClick(TObject *Sender);
729:    void __fastcall butNewReservationClick(TObject *Sender);
730:    void __fastcall butSubmitClick(TObject *Sender);
731:
732:    private: // User declarations
733:
734:    public:     // User declarations
735:       __fastcall TreservationWindow(TComponent* Owner);
736: };
```

```
737: //---------------------------------------------------------------------------
738: extern PACKAGE TreservationWindow *reservationWindow;
739: //---------------------------------------------------------------------------
740: #endif
741:
742:
743: //---------------------------------------------------------------------------
744:
745: #ifndef frmLoginH
746: #define frmLoginH
747: //---------------------------------------------------------------------------
748: #include <Classes.hpp>
749: #include <Controls.hpp>
750: #include <StdCtrls.hpp>
751: #include <Forms.hpp>
752: #include <Db.hpp>
753: #include <DBCtrls.hpp>
754: #include <DBTables.hpp>
755: #include <Mask.hpp>
756: #include <Buttons.hpp>
757: //---------------------------------------------------------------------------
758: class TloginWindow : public TForm
759: {
760:    __published:   // IDE-managed Components
761:       TLabel *Label1;
762:       TLabel *Label2;
763:       TBitBtn *butLogin;
764:       TBitBtn *butCancel;
765:       TEdit *txtUserName;
766:       TEdit *txtPassword;
767:       void __fastcall butLoginClick(TObject *Sender);
768:       void __fastcall butCancelClick(TObject *Sender);
769:
770:    private: // User declarations
771:
772:    public:     // User declarations
773:       __fastcall TloginWindow(TComponent* Owner);
774: };
775: //---------------------------------------------------------------------------
776: extern PACKAGE TloginWindow *loginWindow;
777: //---------------------------------------------------------------------------
778: #endif
779:
780:
781: //---------------------------------------------------------------------------
782:
```

```
783: #ifndef mainWindowH
784: #define mainWindowH
785: //---------------------------------------------------------------------------
786: #include <Classes.hpp>
787: #include <Controls.hpp>
788: #include <StdCtrls.hpp>
789: #include <Forms.hpp>
790: #include <Menus.hpp>
791: #include <ImgList.hpp>
792: #include "dcOutBar.hpp"
793: #include <ComCtrls.hpp>
794: #include "dcOutPanel.hpp"
795:
796: #include "frmLogin.h"
797: #include "frmReservation.h"
798: #include "transactoinWindow.h"
799: #include "frmAccount.h"
800: #include "frmCalendar.h"
801: #include "frmCustomer.h"
802: #include "dataModule.h"
803: #include "frmRoomType.h"
804: #include "frmEmployeeInformation.h"
805: #include "frmEmployeeSchedule.h"
806: #include "frmTimeCard.h"
807: #include "frmFoodService.h"
808: #include "frmRoomInformation.h"
809:
810: //---------------------------------------------------------------------------
811: class TfrmMainWindow : public TForm
812: {
813:    __published:   // IDE-managed Components
814:       TMainMenu *mainMenu;
815:       TMenuItem *mnuFile;
816:       TMenuItem *fileLogin;
817:       TMenuItem *fileLogoff;
818:       TMenuItem *N1;
819:       TMenuItem *fileExit;
820:       TMenuItem *mnuCustomerManagement;
821:       TMenuItem *Reservation1;
822:       TMenuItem *Transaction1;
823:       TMenuItem *mnuServices;
824:       TMenuItem *mnuHelp;
825:       TMenuItem *helpContent;
826:       TMenuItem *N2;
827:       TMenuItem *helpAbout;
828:       TMenuItem *mnuManagement;
```

```
829:       TMenuItem *EmployeeManagement1;
830:       TMenuItem *HotelManagement1;
831:       TMenuItem *RateManagment1;
832:       TMenuItem *ServiceManagement1;
833:       TImageList *imageMainMenu;
834:
835:       TDCOutBar               *dcLeftPanel;
836:       TDCOutBarGroup          *DCOutBarGroup1;
837:       TDCOutBarVertListView   *DCOutBarVertListView1;
838:       TDCOutBarGroup          *DCOutBarGroup2;
839:       TDCOutBarVertListView   *DCOutBarVertListView2;
840:       TDCOutBarGroup          *DCOutBarGroup3;
841:       TDCOutBarVertListView   *DCOutBarVertListView3;
842:
843:       TImageList *imageLeftPanel;
844:       TMenuItem *servicesFood;
845:       TMenuItem *servicesRoom;
846:       TDCOutBarGroup *DCOutBarGroup4;
847:       TDCOutBarVertListView *DCOutBarVertListView4;
848:       TMenuItem *EmployeeInformation1;
849:       TMenuItem *EmployeeSchedule1;
850:       TMenuItem *EmployeeTimeCard1;
851:       TMenuItem *MovieServices1;
852:
853:       void __fastcall fileExitClick(TObject *Sender);
854:       void __fastcall fileLoginClick(TObject *Sender);
855:       void __fastcall Reservation1Click(TObject *Sender);
856:       void __fastcall FormCreate(TObject *Sender);
857:       void __fastcall Transaction1Click(TObject *Sender);
858:
859:       // This is for the Custom Menu Design
860:       void __fastcall MyExpandItemWidth( TObject *Sender, TCanvas *ACanvas, int &Width, int &Height );
861:       void __fastcall MyDrawItem( TObject *Sender, TCanvas *ACanvas, const TRect &ARect, bool Selected );
862:       void __fastcall mnuFileDrawItem(TObject *Sender, TCanvas *ACanvas, TRect &ARect, bool Selected);
863:       void __fastcall mnuFileClick(TObject *Sender);
864:       void __fastcall mnuManagementClick(TObject *Sender);
865:       void __fastcall mnuCustomerManagementClick(TObject *Sender);
866:       void __fastcall mnuServicesClick(TObject *Sender);
867:       void __fastcall mnuHelpClick(TObject *Sender);
868:       void __fastcall fileLogoffClick(TObject *Sender);
869:       void __fastcall helpContentClick(TObject *Sender);
870:       void __fastcall EmployeeInformation1Click(TObject *Sender);
871:       void __fastcall EmployeeSchedule1Click(TObject *Sender);
872:       void __fastcall EmployeeTimeCard1Click(TObject *Sender);
873:       void __fastcall servicesFoodClick(TObject *Sender);
874:       void __fastcall servicesRoomClick(TObject *Sender);
```

```
875:       void __fastcall dcLeftPanelButtonClick(TObject *Sender, TListItem *Item);
876:
877:
878:    private: // User declarations
879:       TColor   MainMenuBackground;
880:       TColor   MainMenuHighlightColor;
881:       TColor   MainMenuTextColor;
882:       TColor   MainMenuTextBackground;
883:       TColor   MainMenuHighlightTextColor;
884:       TColor   Custom;
885:       TColor   VerticalColor;
886:       TColor   MenuColor;
887:       TColor   HighlightColor;
888:       TColor   BorderColor;
889:       TColor   BorderEraseColor;
890:       TColor   NormalTextColor;
891:       TColor   NormalTextBackground;
892:       TColor   HighlightTextColor;
893:       TColor   DisabledTextColor;
894:
895:       int      VerticalWidth;
896:       int      FocusRectRightIndent;
897:       int      FocusRectLeftIndent;
898:       int      LeftTextPos;
899:       int      SideBuffer;
900:       int      MenuIncreaseWidth;
901:       int      Offset;
902:
903:       int      MenuItemHeight;
904:       int      ItemOffset;
905:
906:       TIcon    *Icon;
907:
908:       TreservationWindow   *reservation;
909:       TloginWindow         *login;
910:       TaccountWindow       *account;
911:       TcustomerWindow      *customer;
912:       TtransactionWindow   *transaction;
913:       TroomTypeWindow      *roomType;
914:       TemployeeInformation *employeeInformation;
915:       TemployeeSchedule    *employeeSchedule;
916:       TtimeCard            *timeCard;
917:       TfoodService         *foodService;
918:       TroomInformation     *roomInformation;
919:
920:    public:     // User declarations
```

```
921:
922:         AnsiString   ssn;
923:         AnsiString   startTime;
924:         AnsiString   customerID;
925:         AnsiString   accountID;
926:
927:         __fastcall TfrmMainWindow(TComponent* Owner);
928:
929:         // Functions to handle Window Handle Creation
930:         void __fastcall createReservationWindow( );
931:         void __fastcall createLoginWindow( );
932:         void __fastcall createAccountWindow( );
933:         void __fastcall createCustomerWindow( );
934:         void __fastcall createTransactionWindow( );
935:         void __fastcall createRoomTypeWindow( );
936:         void __fastcall createEmployeeInformationWindow( );
937:         void __fastcall createEmployeeScheduleWindow( );
938:         void __fastcall createTimeCardWindow( );
939:         void __fastcall createFoodServiceWindow( );
940:         void __fastcall createRoomInformationWindow( );
941:
942:         // Functions to handle Window Handle Destruction
943:         void __fastcall destroyReservationWindow( );
944:         void __fastcall destroyLoginWindow( );
945:         void __fastcall destroyAccountWindow( );
946:         void __fastcall destroyCustomerWindow( );
947:         void __fastcall destroyTransactionWindow( );
948:         void __fastcall destroyRoomTypeWindow( );
949:         void __fastcall destroyEmployeeInformationWindow( );
950:         void __fastcall destroyEmployeeScheduleWindow( );
951:         void __fastcall destroyTimeCardWindow( );
952:         void __fastcall destroyFoodServiceWindow( );
953:         void __fastcall destroyRoomInformationWindow( );
954: };
955: //---------------------------------------------------------------------------
956: extern PACKAGE TfrmMainWindow *frmMainWindow;
957: //---------------------------------------------------------------------------
958: #endif
959:
960:
961: //---------------------------------------------------------------------------
962:
963: #include <vcl.h>
964: #pragma hdrstop
965:
966: #include "frmAccount.h"
```

```
 967: #include "mainWindow.h"
 968: //---------------------------------------------------------------------------
 969: #pragma package(smart_init)
 970: #pragma resource "*.dfm"
 971: TaccountWindow *accountWindow;
 972: //---------------------------------------------------------------------------
 973: __fastcall TaccountWindow::TaccountWindow(TComponent* Owner)
 974:    : TForm(Owner)
 975: {
 976: }
 977: //---------------------------------------------------------------------------
 978: void __fastcall TaccountWindow::BitBtn2Click(TObject *Sender)
 979: {
 980:    // Destroy Account Window Handle
 981:    frmMainWindow->destroyAccountWindow( );
 982: }
 983: //---------------------------------------------------------------------------
 984:
 985: void __fastcall TaccountWindow::butNewAccountClick(TObject *Sender)
 986: {
 987:    txtAccountID->Text = "";
 988:    txtCompanyName->Text = "";
 989:    txtCreditCardNumber->Text = "";
 990:    txtExpirationDate->Text = "";
 991:    txtAddress->Text = "";
 992:    txtCity->Text = "";
 993:    txtState->Text = "";
 994:    txtZipCode->Text = "";
 995:    txtBalance->Text = "";
 996:
 997:    txtCompanyName->SetFocus( );
 998: }
 999: //---------------------------------------------------------------------------
1000:
1001: void __fastcall TaccountWindow::butAccountNumberClick(TObject *Sender)
1002: {
1003:    if( !txtAccountID->Text.IsEmpty( ) )
1004:    {
1005:       // search the database for the entered reservation ID
1006:       if( dmHMS->AWcheckAccount( txtAccountID->Text.ToInt() ) )
1007:       {
1008:          txtCompanyName->Text = dmHMS->accountTable->FieldByName("company")->AsString;
1009:          txtCreditCardNumber->Text = dmHMS->accountTable->FieldByName("credit_card_number")->AsString;
1010:          txtExpirationDate->Text = dmHMS->accountTable->FieldByName("expiration_date")->AsString;
1011:          txtAddress->Text = dmHMS->accountTable->FieldByName("billing_address")->AsString;
1012:          txtCity->Text = dmHMS->accountTable->FieldByName("billing_city")->AsString;
```

```
1013:          txtState->Text = dmHMS->accountTable->FieldByName("billing_state")->AsString;
1014:          txtZipCode->Text = dmHMS->accountTable->FieldByName("billing_zip")->AsString;
1015:          txtBalance->Text = dmHMS->accountTable->FieldByName("balance")->AsString;
1016:       }
1017:       else
1018:       {
1019:          // customer not found, clear the fields and prompt again
1020:          txtCompanyName->Text = "";
1021:          txtCreditCardNumber->Text = "";
1022:          txtExpirationDate->Text = "";
1023:          txtAddress->Text = "";
1024:          txtCity->Text = "";
1025:          txtState->Text = "";
1026:          txtZipCode->Text = "";
1027:          txtBalance->Text = "";
1028:
1029:          txtAccountID->SetFocus( );
1030:       }
1031:    }
1032:    else
1033:    {
1034:       // account not found, clear the fields and prompt again
1035:       txtCompanyName->Text = "";
1036:       txtCreditCardNumber->Text = "";
1037:       txtExpirationDate->Text = "";
1038:       txtAddress->Text = "";
1039:       txtCity->Text = "";
1040:       txtState->Text = "";
1041:       txtZipCode->Text = "";
1042:       txtBalance->Text = "";
1043:
1044:       txtAccountID->SetFocus( );
1045:    }
1046: }
1047: //---------------------------------------------------------------------------
1048:
1049: void __fastcall TaccountWindow::butSubmitClick(TObject *Sender)
1050: {
1051:    AnsiString company;
1052:    AnsiString creditCard;
1053:    AnsiString expirationDate;
1054:    AnsiString address;
1055:    AnsiString city;
1056:    AnsiString state;
1057:    AnsiString zip;
1058:
```

```
1059:    // Save account information
1060:    dmHMS->AWsaveAccount( company,
1061:                          creditCard,
1062:                          expirationDate,
1063:                          address,
1064:                          city,
1065:                          state,
1066:                          zip );
1067: }
1068: //---------------------------------------------------------------------------
1069:
1070: //---------------------------------------------------------------------------
1071:
1072: #include <vcl.h>
1073: #pragma hdrstop
1074:
1075: #include "dataModule.h"
1076: //---------------------------------------------------------------------------
1077: #pragma package(smart_init)
1078: #pragma resource "*.dfm"
1079: TdmHMS *dmHMS;
1080: //---------------------------------------------------------------------------
1081: __fastcall TdmHMS::TdmHMS(TComponent* Owner)
1082:    : TDataModule(Owner)
1083: {
1084:    loginTable->Open( );
1085:    timeCardTable->Open( );
1086:    reservationTable->Open( );
1087:    customerTable->Open( );
1088:    accountTable->Open( );
1089:    employeeTable->Open( );
1090:    roomInfoTable->Open( );
1091:    employeeScheduleTable->Open( );
1092:    transactionTable->Open( );
1093: }
1094: //---------------------------------------------------------------------------
1095:
1096: // See if the user exists in the database, return TRUE if so, FALSE otherwise
1097: bool __fastcall TdmHMS::LGcheckUser( AnsiString u, AnsiString p )
1098: {
1099:    dmHMS->loginTable->First( );
1100:    while( !loginTable->Eof )
1101:    {
1102:       if( loginTable->FieldByName("username")->AsString == u &&
1103:           loginTable->FieldByName("password")->AsString == p )
1104:       {
```

```
1105:          return true;
1106:       }
1107:       loginTable->Next( );
1108:    }
1109:    return false;
1110: }
1111:
1112: // Retur ntype of user [Manager, Non-Manager]
1113: int  __fastcall TdmHMS::LGtypeOfUser( AnsiString u )
1114: {
1115:    dmHMS->loginTable->First( );
1116:
1117:    while( !loginTable->Eof )
1118:    {
1119:       if( loginTable->FieldByName("username")->AsString == u )
1120:       {
1121:          return loginTable->FieldByName("security_level")->AsInteger;
1122:       }
1123:       loginTable->Next( );
1124:    }
1125:    return -1;
1126: }
1127:
1128: // Return user Social Security Number
1129: AnsiString __fastcall TdmHMS::LGuserSSN( AnsiString u )
1130: {
1131:    dmHMS->loginTable->First( );
1132:
1133:    while( !loginTable->Eof )
1134:    {
1135:       if( loginTable->FieldByName("username")->AsString == u )
1136:       {
1137:          return loginTable->FieldByName("ssn")->AsString;
1138:       }
1139:       loginTable->Next( );
1140:    }
1141:    return "NotFound";
1142: }
1143:
1144: // Retrieve the start time of an employee
1145: // Remember to update the function to check for the date as well
1146: AnsiString __fastcall TdmHMS::LGstartTime( AnsiString u )
1147: {
1148:    dmHMS->timeCardTable->First( );
1149:
1150:    while( !timeCardTable->Eof )
```

```
1151:    {
1152:       if( timeCardTable->FieldByName("ssn")->AsString == u )
1153:       {
1154:          return timeCardTable->FieldByName("start_time")->AsString;
1155:       }
1156:       timeCardTable->Next( );
1157:    }
1158:    return "NotFound";
1159: }
1160:
1161:
1162: void __fastcall TdmHMS::LGstampTimeCard( AnsiString ssn )
1163: {
1164:    timeCardTable->Insert( );
1165:    timeCardTable->FieldByName("ssn")->Value = ssn;
1166:    timeCardTable->FieldByName("start_time")->Value = TimeToStr( Time( ) );
1167:    timeCardTable->FieldByName("date")->Value = DateToStr( Date( ) );
1168:    timeCardTable->Refresh( );
1169: }
1170:
1171: // Auto time stamp the time card at the end of the session
1172: void __fastcall TdmHMS::MWstampTimeCard( AnsiString ssn, AnsiString startTime )
1173: {
1174:    timeCardTable->Open( );
1175:
1176:    dmHMS->timeCardTable->First( );
1177:
1178:    while( !timeCardTable->Eof )
1179:    {
1180:       if( timeCardTable->FieldByName("ssn")->AsString == ssn &&
1181:           timeCardTable->FieldByName("start_time")->AsString == startTime )
1182:       {
1183:          timeCardTable->Edit( );
1184:          timeCardTable->FieldByName("end_time")->Value = TimeToStr( Time( ) );
1185:          timeCardTable->Post( );
1186:
1187:          return;
1188:       }
1189:       timeCardTable->Next( );
1190:    }
1191:    return;
1192: }
1193:
1194: // Check for reservation
1195: bool __fastcall TdmHMS::RWcheckReservation( int id )
1196: {
```

```
1197:    dmHMS->reservationTable->First( );
1198:    while( !reservationTable->Eof )
1199:    {
1200:       if( reservationTable->FieldByName("reservation_id")->AsInteger == id )
1201:       {
1202:          return true;
1203:       }
1204:       reservationTable->Next( );
1205:    }
1206:    return false;
1207: }
1208:
1209: // Save reservation data
1210: void __fastcall TdmHMS::RWsaveReservation( AnsiString customer,
1211:                                            AnsiString employee,
1212:                                            AnsiString rate,
1213:                                            AnsiString room,
1214:                                            AnsiString start,
1215:                                            AnsiString end )
1216: {
1217:    dmHMS->reservationTable->Last( );
1218:
1219:    int reservation = reservationTable->FieldByName("reservation_id")->AsInteger + 1;
1220:
1221:    reservationTable->Insert( );
1222:    reservationTable->FieldByName("reservation_id")->Value = reservation;
1223:    reservationTable->FieldByName("customer_id")->Value = customer;
1224:    reservationTable->FieldByName("employee_id")->Value = employee;
1225:    reservationTable->FieldByName("rate_id")->Value = rate;
1226:    reservationTable->FieldByName("room_number")->Value = room;
1227:    reservationTable->FieldByName("start_date")->Value = start;
1228:    reservationTable->FieldByName("end_date")->Value = end;
1229:    reservationTable->Refresh( );
1230: }
1231:
1232: // Check customer Information
1233: bool __fastcall TdmHMS::CWcheckCustomer( int id )
1234: {
1235:    dmHMS->customerTable->First( );
1236:    while( !customerTable->Eof )
1237:    {
1238:       if( customerTable->FieldByName("customer_id")->AsInteger == id )
1239:       {
1240:          return true;
1241:       }
1242:       customerTable->Next( );
```

```
1243:    }
1244:    return false;
1245: }
1246:
1247: void __fastcall TdmHMS::CWsaveCustomer(   AnsiString accountID,
1248:                                           AnsiString firstName,
1249:                                           AnsiString lastName,
1250:                                           AnsiString address,
1251:                                           AnsiString city,
1252:                                           AnsiString state,
1253:                                           AnsiString zip,
1254:                                           AnsiString phone )
1255: {
1256:    dmHMS->customerTable->Last( );
1257:
1258:    int customer = customerTable->FieldByName("customer_id")->AsInteger + 1;
1259:
1260:    customerTable->Insert( );
1261:    customerTable->FieldByName("customer_id")->Value = customer;
1262:
1263:    if( !accountID.IsEmpty( ) )
1264:    {
1265:       customerTable->FieldByName("account_id")->Value = accountID.ToInt( );
1266:    }
1267:
1268:    customerTable->FieldByName("address")->Value = address;
1269:    customerTable->FieldByName("phone")->Value = phone;
1270:    customerTable->FieldByName("first_name")->Value = firstName;
1271:    customerTable->FieldByName("last_name")->Value = lastName;
1272:    customerTable->FieldByName("city")->Value = city;
1273:    customerTable->FieldByName("state")->Value = state;
1274:    customerTable->FieldByName("ZipCode")->Value = zip;
1275:    customerTable->Refresh( );
1276: }
1277:
1278: // Check account
1279: bool __fastcall TdmHMS::AWcheckAccount( int id )
1280: {
1281:    dmHMS->accountTable->First( );
1282:    while( !accountTable->Eof )
1283:    {
1284:       if( accountTable->FieldByName("account_id")->AsInteger == id )
1285:       {
1286:          return true;
1287:       }
1288:       accountTable->Next( );
```

```
1289:    }
1290:    return false;
1291: }
1292:
1293: void __fastcall TdmHMS::AWsaveAccount( AnsiString company,
1294:                                        AnsiString creditCard,
1295:                                        AnsiString expiration,
1296:                                        AnsiString address,
1297:                                        AnsiString city,
1298:                                        AnsiString state,
1299:                                        AnsiString zip )
1300: {
1301:    dmHMS->accountTable->Last( );
1302:
1303:    int account = accountTable->FieldByName("account_id")->AsInteger + 1;
1304:
1305:    accountTable->Insert( );
1306:    accountTable->FieldByName("account_id")->Value = account;
1307:    accountTable->FieldByName("credit_card_number")->Value = creditCard.ToInt();
1308:    accountTable->FieldByName("expiration_date")->Value = expiration;
1309:    accountTable->FieldByName("billing_address")->Value = address;
1310:    accountTable->FieldByName("billing_city")->Value = city;
1311:    accountTable->FieldByName("billing_state")->Value = state;
1312:    accountTable->FieldByName("billing_zip")->Value = zip;
1313:    customerTable->Refresh( );
1314: }
1315:
1316: // Check Employee
1317: bool __fastcall TdmHMS::EWcheckEmployee( AnsiString ssn )
1318: {
1319:    dmHMS->employeeTable->First( );
1320:    while( !employeeTable->Eof )
1321:    {
1322:       if( employeeTable->FieldByName("ssn")->AsString == ssn )
1323:       {
1324:          return true;
1325:       }
1326:       employeeTable->Next( );
1327:    }
1328:    return false;
1329: }
1330:
1331: void __fastcall TdmHMS::EWsaveEmployee(  AnsiString ssn,
1332:                                          AnsiString firstName,
1333:                                          AnsiString lastName,
1334:                                          AnsiString address,
```

```
1335:                                          AnsiString city,
1336:                                          AnsiString state,
1337:                                          AnsiString phone,
1338:                                          AnsiString title,
1339:                                          bool salary,
1340:                                          AnsiString rate )
1341: {
1342:    employeeTable->Insert( );
1343:    employeeTable->FieldByName("ssn")->Value = ssn;
1344:    employeeTable->FieldByName("first_name")->Value = firstName;
1345:    employeeTable->FieldByName("last_name")->Value = lastName;
1346:    employeeTable->FieldByName("address")->Value = address;
1347:    employeeTable->FieldByName("city")->Value = city;
1348:    employeeTable->FieldByName("state")->Value = state;
1349:    employeeTable->FieldByName("phone")->Value = phone;
1350:    employeeTable->FieldByName("title")->Value = title;
1351:    employeeTable->FieldByName("salary")->Value = salary;
1352:    employeeTable->FieldByName("rate")->Value = rate.ToInt( );
1353:    employeeTable->Refresh( );
1354: }
1355:
1356: void __fastcall TdmHMS::EWdeleteEmployee( AnsiString ssn )
1357: {
1358:    dmHMS->employeeTable->First( );
1359:    while( !employeeTable->Eof )
1360:    {
1361:       if( employeeTable->FieldByName("ssn")->AsString == ssn )
1362:       {
1363:          employeeTable->Delete( );
1364:          return;
1365:       }
1366:       employeeTable->Next( );
1367:    }
1368:    return;
1369: }
1370:
1371: // Save transaction to the database
1372: void __fastcall TdmHMS::TWsaveTransaction(   AnsiString customer,
1373:                                             AnsiString service,
1374:                                             AnsiString payment,
1375:                                             AnsiString amount )
1376: {
1377:    transactionTable->Insert( );
1378:    transactionTable->FieldByName("customer_id")->Value = customer.ToInt( );
1379:    transactionTable->FieldByName("service_id")->Value = service.ToInt( );
1380:    transactionTable->FieldByName("payment_method")->Value = payment;
```

```
1381:    transactionTable->FieldByName("amount")->Value = amount.ToInt( );
1382:    transactionTable->FieldByName("time")->Value = TimeToStr( Time( ) );
1383:    transactionTable->FieldByName("date")->Value = DateToStr( Date( ) );
1384:    transactionTable->Refresh( );
1385: }
1386:
1387:
1388: //---------------------------------------------------------------------------
1389:
1390: #include <vcl.h>
1391: #pragma hdrstop
1392:
1393: #include "frmRoomInformation.h"
1394: #include "mainWindow.h"
1395: #include "dataModule.h"
1396: //---------------------------------------------------------------------------
1397: #pragma package(smart_init)
1398: #pragma resource "*.dfm"
1399: TroomInformation *roomInformation;
1400: //---------------------------------------------------------------------------
1401: __fastcall TroomInformation::TroomInformation(TComponent* Owner)
1402:    : TForm(Owner)
1403: {
1404: }
1405: //---------------------------------------------------------------------------
1406: void __fastcall TroomInformation::BitBtn1Click(TObject *Sender)
1407: {
1408:    // Destroy Room Information Window Handle
1409:    frmMainWindow->destroyRoomInformationWindow( );
1410: }
1411: //---------------------------------------------------------------------------
1412:
1413: //---------------------------------------------------------------------------
1414:
1415: #include <vcl.h>
1416: #pragma hdrstop
1417:
1418: #include "frmReservation.h"
1419: #include "mainWindow.h"
1420: #include "dataModule.h"
1421: //---------------------------------------------------------------------------
1422: #pragma package(smart_init)
1423: #pragma resource "*.dfm"
1424: TreservationWindow *reservationWindow;
1425: //---------------------------------------------------------------------------
1426: __fastcall TreservationWindow::TreservationWindow(TComponent* Owner)
```

```
1427:    : TForm(Owner)
1428: {
1429: }
1430: //---------------------------------------------------------------------------
1431:
1432: void __fastcall TreservationWindow::butCustomerInformationClick(TObject *Sender)
1433: {
1434:    if( !txtCustomerID->Text.IsEmpty( ) )
1435:    {
1436:       frmMainWindow->customerID = txtCustomerID->Text;
1437:    }
1438:
1439:    // Create Customer Window Handle
1440:    frmMainWindow->createCustomerWindow( );
1441: }
1442: //---------------------------------------------------------------------------
1443:
1444:
1445:
1446: void __fastcall TreservationWindow::butCloseClick(TObject *Sender)
1447: {
1448:    // Destroy Reservation Window Handle
1449:    frmMainWindow->destroyReservationWindow( );
1450: }
1451: //---------------------------------------------------------------------------
1452:
1453:
1454: void __fastcall TreservationWindow::butRateRequestClick(TObject *Sender)
1455: {
1456:    // Create Room Type Window Handle
1457:    frmMainWindow->createRoomTypeWindow( );
1458: }
1459: //---------------------------------------------------------------------------
1460:
1461:
1462: void __fastcall TreservationWindow::butReservationClick(TObject *Sender)
1463: {
1464:    if( txtReservationID->Text != "" )
1465:    {
1466:       // search the database for the entered reservation ID
1467:       if( dmHMS->RWcheckReservation( txtReservationID->Text.ToInt() ) )
1468:       {
1469:          // reservation found will process the information
1470:          txtCustomerID->Text = dmHMS->reservationTable->FieldByName("customer_id")->AsInteger;
1471:          txtRateID->Text = dmHMS->reservationTable->FieldByName("rate_id")->AsInteger;
1472:          txtRoomNumber->Text = dmHMS->reservationTable->FieldByName("room_number")->AsInteger;
```

```
1473:          dtStartDate->Date = dmHMS->reservationTable->FieldByName("start_date")->AsString;
1474:          dtEndDate->Date = dmHMS->reservationTable->FieldByName("end_date")->AsString;
1475:       }
1476:       else
1477:       {
1478:          // reservation not found, clear the fields and prompt again
1479:          txtCustomerID->Text = "";
1480:          txtRateID->Text = "";
1481:          txtRoomNumber->Text = "";
1482:          dtStartDate->Date = Date( );
1483:          dtEndDate->Date = Date( );
1484:
1485:          txtReservationID->SetFocus( );
1486:       }
1487:    }
1488:    else
1489:    {
1490:       // reservation not found, clear the fields and prompt again
1491:       txtCustomerID->Text = "";
1492:       txtRateID->Text = "";
1493:       txtRoomNumber->Text = "";
1494:       dtStartDate->Date = Date( );
1495:       dtEndDate->Date = Date( );
1496:
1497:       txtReservationID->SetFocus( );
1498:    }
1499: }
1500: //---------------------------------------------------------------------------
1501:
1502: void __fastcall TreservationWindow::butNewReservationClick(TObject *Sender)
1503: {
1504:    txtReservationID->Text = "";
1505:    txtCustomerID->Text = "";
1506:    txtRateID->Text = "";
1507:    txtRoomNumber->Text = "";
1508:    dtStartDate->Date = Date( );
1509:    dtEndDate->Date = Date( );
1510:
1511:    // code here to open database
1512:    txtCustomerID->SetFocus( );
1513: }
1514: //---------------------------------------------------------------------------
1515:
1516: void __fastcall TreservationWindow::butSubmitClick(TObject *Sender)
1517: {
1518:    AnsiString customerID = txtCustomerID->Text;
```

```
1519:    AnsiString employeeID = frmMainWindow->ssn;
1520:    AnsiString rateID = txtRateID->Text;
1521:    AnsiString roomNumber = txtRoomNumber->Text;
1522:    AnsiString startDate = dtStartDate->Date.DateString( );
1523:    AnsiString endDate = dtEndDate->Date.DateString( );
1524:
1525:    // save to the database
1526:    dmHMS->RWsaveReservation(  customerID,
1527:                               employeeID,
1528:                               rateID,
1529:                               roomNumber,
1530:                               startDate,
1531:                               endDate );
1532:
1533: }
1534: //---------------------------------------------------------------------------
1535:
1536: //---------------------------------------------------------------------------
1537:
1538: #include <vcl.h>
1539: #pragma hdrstop
1540:
1541: #include "frmLogin.h"
1542: #include "dataModule.h"
1543: #include "mainWindow.h"
1544:
1545: //---------------------------------------------------------------------------
1546: #pragma package(smart_init)
1547: #pragma resource "*.dfm"
1548: TloginWindow *loginWindow;
1549: //---------------------------------------------------------------------------
1550: __fastcall TloginWindow::TloginWindow(TComponent* Owner)
1551:         : TForm(Owner)
1552: {
1553: }
1554: //---------------------------------------------------------------------------
1555:
1556:
1557: void __fastcall TloginWindow::butLoginClick(TObject *Sender)
1558: {
1559:    // Check the database
1560:    if( dmHMS->LGcheckUser( txtUserName->Text, txtPassword->Text ) )
1561:    {
1562:       // Determine User Type
1563:       switch( dmHMS->LGtypeOfUser( txtUserName->Text ) )
1564:       {
```

```
1565:          case 0:     // Not a manager
1566:          {
1567:             // ---------------------------------------------------------------------
1568:             // Enable/Disable Appropriate menus
1569:             // ---------------------------------------------------------------------
1570:             frmMainWindow->mnuCustomerManagement->Enabled = true;
1571:             frmMainWindow->mnuServices->Enabled = true;
1572:
1573:             frmMainWindow->fileLogoff->Enabled = true;
1574:             frmMainWindow->fileLogin->Enabled = false;
1575:
1576:             frmMainWindow->dcLeftPanel->Groups[0]->Enabled = true;
1577:             frmMainWindow->dcLeftPanel->Groups[1]->Enabled = true;
1578:             frmMainWindow->dcLeftPanel->Groups[3]->Enabled = true;
1579:             // ---------------------------------------------------------------------
1580:
1581:             break;
1582:          }
1583:
1584:          case 1:     // Manager
1585:          {
1586:             // ---------------------------------------------------------------------
1587:             // Enable/Disable Appropriate menus
1588:             // ---------------------------------------------------------------------
1589:             frmMainWindow->mnuManagement->Enabled = true;
1590:             frmMainWindow->mnuCustomerManagement->Enabled = true;
1591:             frmMainWindow->mnuServices->Enabled = true;
1592:
1593:             frmMainWindow->fileLogoff->Enabled = true;
1594:             frmMainWindow->fileLogin->Enabled = false;
1595:
1596:             frmMainWindow->dcLeftPanel->Groups[0]->Enabled = true;
1597:             frmMainWindow->dcLeftPanel->Groups[1]->Enabled = true;
1598:             frmMainWindow->dcLeftPanel->Groups[2]->Enabled = true;
1599:             frmMainWindow->dcLeftPanel->Groups[3]->Enabled = true;
1600:             // ---------------------------------------------------------------------
1601:
1602:             break;
1603:          }
1604:
1605:          default :
1606:          {
1607:             txtPassword->Text = "";
1608:             txtUserName->Text = "";
1609:             txtUserName->SetFocus( );
1610:
```

```
1611:             break;
1612:          }
1613:       }
1614:
1615:       // Stamp the time card automatically
1616:       AnsiString Social = dmHMS->LGuserSSN( txtUserName->Text );
1617:       dmHMS->LGstampTimeCard( Social );
1618:
1619:       AnsiString StartTime = dmHMS->LGstartTime( Social );
1620:
1621:       // Destroy Login Window Handle
1622:       frmMainWindow->ssn = Social;
1623:       frmMainWindow->startTime = StartTime;
1624:
1625:       frmMainWindow->destroyLoginWindow( );
1626:    }
1627:    else
1628:    {
1629:       txtPassword->Text = "";
1630:       txtUserName->Text = "";
1631:       txtUserName->SetFocus( );
1632:    }
1633: }
1634: //---------------------------------------------------------------------------
1635:
1636: void __fastcall TloginWindow::butCancelClick(TObject *Sender)
1637: {
1638:    // Destroy Login Window Handle
1639:    frmMainWindow->destroyLoginWindow( );
1640: }
1641: //---------------------------------------------------------------------------
1642:
1643:
1644:
1645: //---------------------------------------------------------------------------
1646:
1647: #include <vcl.h>
1648: #pragma hdrstop
1649:
1650: #include "frmFoodService.h"
1651: #include "mainWindow.h"
1652: //---------------------------------------------------------------------------
1653: #pragma package(smart_init)
1654: #pragma resource "*.dfm"
1655: TfoodService *foodService;
1656: //---------------------------------------------------------------------------
```

```
1657: __fastcall TfoodService::TfoodService(TComponent* Owner)
1658:    : TForm(Owner)
1659: {
1660: }
1661: //---------------------------------------------------------------------------
1662: void __fastcall TfoodService::BitBtn1Click(TObject *Sender)
1663: {
1664:    // Destroy Food Service Window Handle
1665:    frmMainWindow->destroyFoodServiceWindow( );
1666: }
1667: //---------------------------------------------------------------------------
1668:
1669: //---------------------------------------------------------------------------
1670:
1671: #include <vcl.h>
1672: #pragma hdrstop
1673:
1674: #include "frmEmployeeSchedule.h"
1675: #include "mainWindow.h"
1676: #include "dataModule.h"
1677: //---------------------------------------------------------------------------
1678: #pragma package(smart_init)
1679: #pragma resource "*.dfm"
1680: TemployeeSchedule *employeeSchedule;
1681: //---------------------------------------------------------------------------
1682: __fastcall TemployeeSchedule::TemployeeSchedule(TComponent* Owner)
1683:    : TForm(Owner)
1684: {
1685: }
1686: //---------------------------------------------------------------------------
1687: void __fastcall TemployeeSchedule::butCloseClick(TObject *Sender)
1688: {
1689:    // Destroy Employee Schedule Window Handle
1690:    frmMainWindow->destroyEmployeeScheduleWindow( );
1691: }
1692: //---------------------------------------------------------------------------
1693:
1694:
1695: //---------------------------------------------------------------------------
1696:
1697: #include <vcl.h>
1698: #pragma hdrstop
1699:
1700: #include "frmEmployeeInformation.h"
1701: #include "mainWindow.h"
1702: //---------------------------------------------------------------------------
```

```
1703: #pragma package(smart_init)
1704: #pragma resource "*.dfm"
1705: TemployeeInformation *employeeInformation;
1706: //---------------------------------------------------------------------------
1707: __fastcall TemployeeInformation::TemployeeInformation(TComponent* Owner)
1708:    : TForm(Owner)
1709: {
1710: }
1711: //---------------------------------------------------------------------------
1712: void __fastcall TemployeeInformation::butCloseClick(TObject *Sender)
1713: {
1714:    // Destroy Employee Information Winodw
1715:    frmMainWindow->destroyEmployeeInformationWindow( );
1716: }
1717: //---------------------------------------------------------------------------
1718: void __fastcall TemployeeInformation::butNewCustomerClick(TObject *Sender)
1719: {
1720:    txtSSN->Text = "";
1721:    txtFirstName->Text = "";
1722:    txtLastName->Text = "";
1723:    txtAddress->Text = "";
1724:    txtCity->Text = "";
1725:    txtState->Text = "";
1726:    txtPhone->Text = "";
1727:    txtTitle->Text = "";
1728:    cbSalary->Checked = false;
1729:    txtRate->Text = "";
1730:
1731:    butSubmit->Enabled = true;
1732: }
1733: //---------------------------------------------------------------------------
1734:
1735: void __fastcall TemployeeInformation::butSSNClick(TObject *Sender)
1736: {
1737:    if( !txtSSN->Text.IsEmpty( ) )
1738:    {
1739:       // search the database for the entered Social Security Number
1740:       if( dmHMS->EWcheckEmployee( txtSSN->Text ) )
1741:       {
1742:          txtFirstName->Text = dmHMS->employeeTable->FieldByName("first_name")->AsString;
1743:          txtLastName->Text = dmHMS->employeeTable->FieldByName("last_name")->AsString;
1744:          txtAddress->Text = dmHMS->employeeTable->FieldByName("address")->AsString;
1745:          txtCity->Text = dmHMS->employeeTable->FieldByName("city")->AsString;
1746:          txtState->Text = dmHMS->employeeTable->FieldByName("state")->AsString;
1747:          txtPhone->Text = dmHMS->employeeTable->FieldByName("phone")->AsString;
1748:          txtTitle->Text = dmHMS->employeeTable->FieldByName("title")->AsString;
```

```
1749:          cbSalary->Checked = dmHMS->employeeTable->FieldByName("salary")->AsBoolean;
1750:          txtRate->Text = dmHMS->employeeTable->FieldByName("rate")->AsString;
1751:
1752:          butDelete->Enabled = true;
1753:       }
1754:       else
1755:       {
1756:          // customer not found, clear the fields and prompt again
1757:          txtSSN->Text = "";
1758:          txtFirstName->Text = "";
1759:          txtLastName->Text = "";
1760:          txtAddress->Text = "";
1761:          txtCity->Text = "";
1762:          txtState->Text = "";
1763:          txtPhone->Text = "";
1764:          txtTitle->Text = "";
1765:          cbSalary->Checked = false;
1766:          txtRate->Text = "";
1767:
1768:          txtSSN->SetFocus( );
1769:       }
1770:    }
1771:    else
1772:    {
1773:       // customer not found, clear the fields and prompt again
1774:          txtSSN->Text = "";
1775:          txtFirstName->Text = "";
1776:          txtLastName->Text = "";
1777:          txtAddress->Text = "";
1778:          txtCity->Text = "";
1779:          txtState->Text = "";
1780:          txtPhone->Text = "";
1781:          txtTitle->Text = "";
1782:          cbSalary->Checked = false;
1783:          txtRate->Text = "";
1784:
1785:          txtSSN->SetFocus( );
1786:    }
1787: }
1788: //---------------------------------------------------------------------------
1789:
1790: void __fastcall TemployeeInformation::butSubmitClick(TObject *Sender)
1791: {
1792:    AnsiString ssn = txtSSN->Text;
1793:    AnsiString firstName = txtFirstName->Text;
1794:    AnsiString lastName = txtLastName->Text;
```

```
1795:    AnsiString address = txtAddress->Text;
1796:    AnsiString city = txtCity->Text;
1797:    AnsiString state = txtState->Text;
1798:    AnsiString phone = txtPhone->Text;
1799:    AnsiString title = txtTitle->Text;
1800:    bool       salary = cbSalary->Checked;
1801:    AnsiString rate = txtRate->Text;
1802:
1803:    // Save information to the database
1804:    dmHMS->EWsaveEmployee( ssn,
1805:                           firstName,
1806:                           lastName,
1807:                           address,
1808:                           city,
1809:                           state,
1810:                           phone,
1811:                           title,
1812:                           salary,
1813:                           rate );
1814:
1815:    butSubmit->Enabled = false;
1816: }
1817: //---------------------------------------------------------------------------
1818:
1819:
1820: void __fastcall TemployeeInformation::butDeleteClick(TObject *Sender)
1821: {
1822:    dmHMS->EWdeleteEmployee( txtSSN->Text );
1823:
1824:    txtSSN->Text = "";
1825:    txtFirstName->Text = "";
1826:    txtLastName->Text = "";
1827:    txtAddress->Text = "";
1828:    txtCity->Text = "";
1829:    txtState->Text = "";
1830:    txtPhone->Text = "";
1831:    txtTitle->Text = "";
1832:    cbSalary->Checked = false;
1833:    txtRate->Text = "";
1834:
1835:    txtSSN->SetFocus( );
1836:
1837:    butDelete->Enabled = false;
1838: }
1839: //---------------------------------------------------------------------------
1840:
```

```
1841: void __fastcall TemployeeInformation::FormClose(TObject *Sender,
1842:       TCloseAction &Action)
1843: {
1844:    txtSSN->Text = "";
1845:    txtFirstName->Text = "";
1846:    txtLastName->Text = "";
1847:    txtAddress->Text = "";
1848:    txtCity->Text = "";
1849:    txtState->Text = "";
1850:    txtPhone->Text = "";
1851:    txtTitle->Text = "";
1852:    cbSalary->Checked = false;
1853:    txtRate->Text = "";
1854: }
1855: //---------------------------------------------------------------------------
1856:
1857:
1858: //---------------------------------------------------------------------------
1859:
1860: #include <vcl.h>
1861: #pragma hdrstop
1862:
1863: #include "frmCustomer.h"
1864: #include "frmReservation.h"
1865: #include "mainWindow.h"
1866: //---------------------------------------------------------------------------
1867: #pragma package(smart_init)
1868: #pragma resource "*.dfm"
1869: TcustomerWindow *customerWindow;
1870: //---------------------------------------------------------------------------
1871: __fastcall TcustomerWindow::TcustomerWindow(TComponent* Owner)
1872:    : TForm(Owner)
1873: {
1874: }
1875: //---------------------------------------------------------------------------
1876:
1877: void __fastcall TcustomerWindow::butCloseClick(TObject *Sender)
1878: {
1879:    // Destroy Customer Window Handle
1880:    frmMainWindow->destroyCustomerWindow( );
1881:
1882:    // might need to add focus to customer window
1883: }
1884: //---------------------------------------------------------------------------
1885:
1886: void __fastcall TcustomerWindow::butAccountIDClick(TObject *Sender)
```

```
1887: {
1888:    if( !txtAccountID->Text.IsEmpty( ) )
1889:    {
1890:       frmMainWindow->accountID = txtAccountID->Text;
1891:    }
1892:
1893:    // Create Account Window Handle
1894:    frmMainWindow->createAccountWindow( );
1895: }
1896: //---------------------------------------------------------------------------
1897:
1898: void __fastcall TcustomerWindow::butCustomerIDClick(TObject *Sender)
1899: {
1900:    if( txtCustomerID->Text != "" )
1901:    {
1902:       // search the database for the entered reservation ID
1903:       if( dmHMS->CWcheckCustomer( txtCustomerID->Text.ToInt() ) )
1904:       {
1905:          txtAccountID->Text = dmHMS->customerTable->FieldByName("account_id")->AsInteger;
1906:          txtFirstName->Text = dmHMS->customerTable->FieldByName("first_name")->AsString;
1907:          txtLastName->Text = dmHMS->customerTable->FieldByName("last_name")->AsString;
1908:          txtAddress->Text = dmHMS->customerTable->FieldByName("address")->AsString;
1909:          txtCity->Text = dmHMS->customerTable->FieldByName("city")->AsString;
1910:          txtState->Text = dmHMS->customerTable->FieldByName("state")->AsString;
1911:          txtZipCode->Text = dmHMS->customerTable->FieldByName("ZipCode")->AsString;
1912:          txtPhone->Text = dmHMS->customerTable->FieldByName("phone")->AsString;
1913:       }
1914:       else
1915:       {
1916:          // customer not found, clear the fields and prompt again
1917:          txtAccountID->Text = "";
1918:          txtFirstName->Text = "";
1919:          txtLastName->Text = "";
1920:          txtAddress->Text = "";
1921:          txtCity->Text = "";
1922:          txtState->Text = "";
1923:          txtZipCode->Text = "";
1924:          txtPhone->Text = "";
1925:
1926:          txtCustomerID->SetFocus( );
1927:       }
1928:    }
1929:    else
1930:    {
1931:       // customer not found, clear the fields and prompt again
1932:       txtAccountID->Text = "";
```

```
1933:       txtFirstName->Text = "";
1934:       txtLastName->Text = "";
1935:       txtAddress->Text = "";
1936:       txtCity->Text = "";
1937:       txtState->Text = "";
1938:       txtZipCode->Text = "";
1939:       txtPhone->Text = "";
1940:
1941:       txtCustomerID->SetFocus( );
1942:    }
1943: }
1944: //---------------------------------------------------------------------------
1945:
1946: void __fastcall TcustomerWindow::butNewCustomerClick(TObject *Sender)
1947: {
1948:    txtCustomerID->Text = "";
1949:    txtAccountID->Text = "";
1950:    txtFirstName->Text = "";
1951:    txtLastName->Text = "";
1952:    txtAddress->Text = "";
1953:    txtCity->Text = "";
1954:    txtState->Text = "";
1955:    txtZipCode->Text = "";
1956:    txtPhone->Text = "";
1957:
1958:    txtFirstName->SetFocus( );
1959: }
1960: //---------------------------------------------------------------------------
1961:
1962: void __fastcall TcustomerWindow::butSubmitClick(TObject *Sender)
1963: {
1964:    AnsiString accountID = txtAccountID->Text;
1965:    AnsiString firstName = txtFirstName->Text;
1966:    AnsiString lastName = txtLastName->Text;
1967:    AnsiString address = txtAddress->Text;
1968:    AnsiString city = txtCity->Text;
1969:    AnsiString state = txtState->Text;
1970:    AnsiString zip = txtZipCode->Text;
1971:    AnsiString phone = txtPhone->Text;
1972:
1973:    // Save to the database
1974:    dmHMS->CWsaveCustomer( accountID,
1975:                           firstName,
1976:                           lastName,
1977:                           address,
1978:                           city,
```

```
1979:                          state,
1980:                          zip,
1981:                          phone );
1982: }
1983: //---------------------------------------------------------------------------
1984:
1985: //---------------------------------------------------------------------------
1986:
1987: #include <vcl.h>
1988: #pragma hdrstop
1989:
1990: #include "frmTimeCard.h"
1991: #include "mainWindow.h"
1992: #include "dataModule.h"
1993: //---------------------------------------------------------------------------
1994: #pragma package(smart_init)
1995: #pragma resource "*.dfm"
1996: TtimeCard *timeCard;
1997: //---------------------------------------------------------------------------
1998: __fastcall TtimeCard::TtimeCard(TComponent* Owner)
1999:    : TForm(Owner)
2000: {
2001: }
2002: //---------------------------------------------------------------------------
2003: void __fastcall TtimeCard::butCloseClick(TObject *Sender)
2004: {
2005:    frmMainWindow->destroyTimeCardWindow( );
2006: }
2007: //---------------------------------------------------------------------------
2008:
2009: //---------------------------------------------------------------------------
2010:
2011: #include <vcl.h>
2012: #pragma hdrstop
2013:
2014: #include "frmRoomType.h"
2015: #include "mainWindow.h"
2016: //---------------------------------------------------------------------------
2017: #pragma package(smart_init)
2018: #pragma resource "*.dfm"
2019: TroomTypeWindow *roomTypeWindow;
2020: //---------------------------------------------------------------------------
2021: __fastcall TroomTypeWindow::TroomTypeWindow(TComponent* Owner)
2022:    : TForm(Owner)
2023: {
2024: }
```

```
2025: //---------------------------------------------------------------------------
2026: void __fastcall TroomTypeWindow::butCloseClick(TObject *Sender)
2027: {
2028:    // Destroy Room Type Window Handle
2029:    frmMainWindow->destroyRoomTypeWindow( );
2030: }
2031: //---------------------------------------------------------------------------
2032:
2033:
2034: //---------------------------------------------------------------------------
2035:
2036: #include <vcl.h>
2037: #pragma hdrstop
2038:
2039: #include "transactoinWindow.h"
2040: #include "mainWindow.h"
2041: #include "dataModule.h"
2042: //---------------------------------------------------------------------------
2043: #pragma package(smart_init)
2044: #pragma resource "*.dfm"
2045: TtransactionWindow *transactionWindow;
2046: //---------------------------------------------------------------------------
2047: __fastcall TtransactionWindow::TtransactionWindow(TComponent* Owner)
2048:    : TForm(Owner)
2049: {
2050: }
2051: //---------------------------------------------------------------------------
2052: void __fastcall TtransactionWindow::butCloseClick(TObject *Sender)
2053: {
2054:    // Destroy Transaction Window Handle
2055:    frmMainWindow->destroyTransactionWindow( );
2056: }
2057: //---------------------------------------------------------------------------
2058:
2059: void __fastcall TtransactionWindow::butSubmitClick(TObject *Sender)
2060: {
2061:    AnsiString customer = txtCustomerID->Text;
2062:    AnsiString service = txtServiceID->Text;
2063:    AnsiString payment = cbPaymentMethod->Text;
2064:    AnsiString amount = txtAmountDue->Text;
2065:
2066:    // save to the database
2067:    dmHMS->TWsaveTransaction(  customer,
2068:                               service,
2069:                               payment,
2070:                               amount );
```

```
2071: }
2072: //---------------------------------------------------------------------------
2073:
2074: //---------------------------------------------------------------------------
2075:
2076: #include <vcl.h>
2077: #pragma hdrstop
2078:
2079: #include "mainWindow.h"
2080:
2081: //---------------------------------------------------------------------------
2082: #pragma package(smart_init)
2083: #pragma link "dcOutBar"
2084: #pragma link "dcOutPanel"
2085: #pragma resource "*.dfm"
2086: TfrmMainWindow *frmMainWindow;
2087:
2088: const AnsiString BLANK_LINE = "-";
2089:
2090: //---------------------------------------------------------------------------
2091: __fastcall TfrmMainWindow::TfrmMainWindow(TComponent* Owner)
2092:         : TForm(Owner)
2093: {
2094: }
2095: //---------------------------------------------------------------------------
2096:
2097: void __fastcall TfrmMainWindow::fileExitClick(TObject *Sender)
2098: {
2099:    Close( ); // Terminate program
2100: }
2101: //---------------------------------------------------------------------------
2102:
2103: void __fastcall TfrmMainWindow::fileLoginClick(TObject *Sender)
2104: {
2105:    // Create Login Window Handle
2106:    createLoginWindow( );
2107: }
2108: //---------------------------------------------------------------------------
2109:
2110: void __fastcall TfrmMainWindow::Reservation1Click(TObject *Sender)
2111: {
2112:    // Create Reservation Window
2113:    createReservationWindow( );
2114: }
2115: //---------------------------------------------------------------------------
2116:
```

```
2117: void __fastcall TfrmMainWindow::FormCreate(TObject *Sender)
2118: {
2119:    this->Caption = "Hotel Information Management System";  // Window Title
2120:    this->WindowState = wsMaximized;                        // Maximize Window
2121:
2122:    // Hotel Management System Help File
2123:    Application->HelpFile = "HIMS.HLP";
2124:
2125:
2126:    // Custom Outlook Like Bar Design
2127:    dcLeftPanel->ActiveGroupIndex = 0;
2128:
2129:    dcLeftPanel->Groups[0]->Caption = "Reservation";
2130:    dcLeftPanel->Groups[1]->Caption = "Hotel Services";
2131:    dcLeftPanel->Groups[2]->Caption = "Management";
2132:    dcLeftPanel->Groups[3]->Caption = "Employee";
2133:
2134:    // Disable Left Panel
2135:    dcLeftPanel->Groups[0]->Enabled = false;
2136:    dcLeftPanel->Groups[1]->Enabled = false;
2137:    dcLeftPanel->Groups[2]->Enabled = false;
2138:    dcLeftPanel->Groups[3]->Enabled = false;
2139:
2140:    // Custom Design Menu System
2141:    // ---------------------------------------------------------------------
2142:
2143:    // Custom light blue color
2144:    Custom=TColor(RGB(185,239,245));
2145:
2146:    // Used to draw and highlight the main menu items
2147:    MainMenuBackground = clBtnFace;
2148:    MainMenuHighlightColor = Custom;
2149:    MainMenuTextColor = clBlack;
2150:    MainMenuTextBackground = clSilver;
2151:    MainMenuHighlightTextColor = clRed;
2152:
2153:    // Used to draw and color the menu items
2154:    VerticalColor = clSilver;
2155:    MenuColor = clWhite;
2156:    HighlightColor = Custom;
2157:    BorderColor = clBlack;
2158:    NormalTextColor = clBlack;
2159:    NormalTextBackground = clWhite;
2160:    HighlightTextColor = clBlack;
2161:    DisabledTextColor = clSilver;
2162:
```

```
2163:    // Width of the vertical bar on the right
2164:    VerticalWidth = 26;
2165:
2166:    // Space buffer between the sides of the menu
2167:    // and the sides of the focus rect
2168:    FocusRectRightIndent = 3;
2169:    FocusRectLeftIndent = 3;
2170:
2171:    // Position to draw the text
2172:    LeftTextPos = 35;
2173:
2174:    // Space between the focus rect outline and the focus rect
2175:    SideBuffer = 1;
2176:
2177:    // Assign an image list makes the width wider, so we need to adjust when
2178:    // you do and don't have one.
2179:    if(Menu->Images == NULL)
2180:       MenuIncreaseWidth = 100;
2181:    else
2182:       MenuIncreaseWidth = 50;
2183:
2184:    // Offset will increase the Height of the menu items.
2185:    // Also used to make sure the icons are aligned correctly.
2186:    Offset = 5;
2187:    // ---------------------------------------------------------------------
2188:
2189:    // ---------------------------------------------------------------------
2190:    // Diable Menu System
2191:    // ---------------------------------------------------------------------
2192:    mnuManagement->Enabled = false;
2193:    mnuCustomerManagement->Enabled = false;
2194:    mnuServices->Enabled = false;
2195:
2196:    fileLogoff->Enabled = false;
2197:    // ---------------------------------------------------------------------
2198: }
2199:
2200: // ---------------------------------------------------------------------------
2201: void __fastcall TfrmMainWindow::Transaction1Click(TObject *Sender)
2202: {
2203:    // Create Transaction Window Handle
2204:    createTransactionWindow( );
2205: }
2206: //---------------------------------------------------------------------------
2207:
2208: //---------------------------------------------------------------------------
```

```
2209: void __fastcall TfrmMainWindow::mnuFileDrawItem( TObject *Sender,
2210:                                                  TCanvas *ACanvas,
2211:                                                  TRect &ARect,
2212:                                                  bool Selected)
2213: {
2214:    TRect FocusRectBorder;
2215:    TRect FocusRectFill;
2216:    TMenuItem *MenuItem = ((TMenuItem*)Sender);
2217:
2218:    AnsiString Text = MenuItem->Caption;
2219:
2220:    // Color the background behind the text
2221:    ACanvas->Brush->Color = MainMenuBackground;
2222:    ACanvas->FillRect(ARect);
2223:
2224:    // For any blank items acting as seperators
2225:    if(Text == "")
2226:       return;
2227:
2228:    if(Selected)
2229:    {
2230:       // Draw the outline of the selection box
2231:       FocusRectBorder = ARect;
2232:       ACanvas->Brush->Color = BorderColor;
2233:       ACanvas->FrameRect(FocusRectBorder);
2234:
2235:
2236:       // Draw the inside of the selection box
2237:       // Make is a little smaller so it does not erase the outline
2238:       FocusRectFill = ARect;
2239:       FocusRectFill.Top += SideBuffer;
2240:       FocusRectFill.Right -= SideBuffer;
2241:       FocusRectFill.Left += SideBuffer;
2242:       FocusRectFill.Bottom -= SideBuffer;
2243:       ACanvas->Brush->Color = MainMenuHighlightColor;
2244:       ACanvas->FillRect(FocusRectFill);
2245:
2246:       // Set the color that we want our text drawn when the item is selected
2247:       ACanvas->Font->Color = MainMenuHighlightTextColor;
2248:    }
2249:    else
2250:    {
2251:       ACanvas->Font->Color = MainMenuTextColor;
2252:    }
2253:
2254:
```

```
2255:    int TextLength;
2256:    TRect TextRect;
2257:
2258:    TextLength = Text.Length();
2259:    TextRect = ARect;
2260:    // This determines where the text is drawn.
2261:    TextRect.Left += 5;
2262:    TextRect.Top += 1;
2263:
2264:    // Draw the text
2265:    DrawText(ACanvas->Handle,Text.c_str(), TextLength, &TextRect, 0);
2266:
2267: }
2268: //---------------------------------------------------------------------------
2269:
2270:
2271: //---------------------------------------------------------------------------
2272: void __fastcall TfrmMainWindow::MyExpandItemWidth(TObject *Sender,
2273:    TCanvas *ACanvas, int &Width, int &Height)
2274: {
2275:    Width += MenuIncreaseWidth;
2276:    Height += Offset;
2277:    MenuItemHeight = Height;
2278:    ItemOffset = Offset/2;
2279: }
2280: //---------------------------------------------------------------------------
2281:
2282: //---------------------------------------------------------------------------
2283: void __fastcall TfrmMainWindow::MyDrawItem(TObject* Sender, TCanvas* ACanvas,
2284:    const TRect &ARect, bool Selected)
2285: {
2286:    int TopPos, TextLength;
2287:    AnsiString Text;
2288:    TRect TempRect;
2289:    TRect VerticalRect;
2290:    TRect FocusRectBorder;
2291:    TRect FocusRectFill;
2292:    TRect TextRect;
2293:
2294:    TMenuItem *MenuItem = ((TMenuItem*)Sender);
2295:
2296:    Text = MenuItem->Caption;
2297:
2298:    // Draw the Background erases anything that was there before.
2299:    ACanvas->Brush->Color = MenuColor;
2300:    ACanvas->FillRect(ARect);
```

```
2301:
2302:    // Draw any seperator lines
2303:    if(Text==BLANK_LINE)
2304:    {
2305:       // Draw the Vertical Bar
2306:       VerticalRect = ARect;
2307:       VerticalRect.Top -= SideBuffer;
2308:       VerticalRect.Right = VerticalWidth;
2309:       VerticalRect.Bottom += SideBuffer;
2310:       ACanvas->Brush->Color = VerticalColor;
2311:       ACanvas->FillRect(VerticalRect);
2312:
2313:       // Draw the Blank Line
2314:       ACanvas->MoveTo(VerticalWidth,ARect.Top+ARect.Height()/2);
2315:       ACanvas->LineTo(ARect.Right,ARect.Top+ARect.Height()/2);
2316:       return;
2317:    }
2318:
2319:    // This is for non seperator lines
2320:
2321:    TextLength = Text.Length();
2322:
2323:    if(Selected)
2324:    {
2325:       // Have to draw the vertical bar section to fill in the area that is not
2326:       // covered by the selection rect
2327:       VerticalRect = ARect;
2328:       VerticalRect.Top -= SideBuffer;
2329:       VerticalRect.Right = VerticalWidth;
2330:       VerticalRect.Bottom += SideBuffer;
2331:       ACanvas->Brush->Color = VerticalColor;
2332:       ACanvas->FillRect(VerticalRect);
2333:
2334:       if(MenuItem->Enabled)
2335:       {
2336:          // The item is selected and enabled
2337:
2338:          // Draw the focus rect outline border
2339:          FocusRectBorder = ARect;
2340:          FocusRectBorder.Left += FocusRectLeftIndent - SideBuffer;
2341:          FocusRectBorder.Right -= FocusRectRightIndent - SideBuffer;
2342:          ACanvas->Brush->Color = BorderColor;
2343:          ACanvas->FrameRect(FocusRectBorder);
2344:
2345:
2346:          // Fill in the focus rect.  Making it a little smaller so as to not
```

```
2347:          // draw over the outline
2348:          FocusRectFill = ARect;
2349:          FocusRectFill.Right -= FocusRectRightIndent;
2350:          FocusRectFill.Left += FocusRectLeftIndent;
2351:          FocusRectFill.Bottom -= SideBuffer;
2352:          FocusRectFill.Top += SideBuffer;
2353:          ACanvas->Brush->Color = HighlightColor;
2354:          ACanvas->FillRect(FocusRectFill);
2355:
2356:          // Set the way we want to draw our text
2357:          ACanvas->Font->Color = HighlightTextColor;
2358:          ACanvas->Font->Style = TFontStyles() << fsBold;
2359:       }
2360:       else
2361:       {
2362:          // Menu item is not enabled so do not draw any selection
2363:          // rect and change the text color to reflect its unenabled state
2364:          // NOTE: The icon is still drawn normally and not disabled
2365:          ACanvas->Font->Style = TFontStyles();
2366:          ACanvas->Brush->Color = NormalTextBackground;
2367:          ACanvas->Font->Color = DisabledTextColor;
2368:       }
2369:
2370:    }
2371:    else
2372:    {
2373:       // Fill in the vertical area
2374:       VerticalRect = ARect;
2375:       VerticalRect.Top -= SideBuffer;
2376:       VerticalRect.Right = VerticalWidth;
2377:       VerticalRect.Bottom += SideBuffer;
2378:       ACanvas->Brush->Color = VerticalColor;
2379:       ACanvas->FillRect(VerticalRect);
2380:
2381:       // Set the text background color and font based on if it is enabled or not
2382:
2383:       if(MenuItem->Enabled)
2384:       {
2385:          ACanvas->Brush->Color = NormalTextBackground;
2386:          ACanvas->Font->Color = NormalTextColor;
2387:       }
2388:       else
2389:       {
2390:          ACanvas->Brush->Color = NormalTextBackground;
2391:          ACanvas->Font->Color = DisabledTextColor;
2392:       }
```

```
2393:
2394:    }
2395:
2396:    // Calculate out the Rect we want to draw our text in
2397:    TextRect = ARect;
2398:    TextRect.Left += LeftTextPos;
2399:    if(Offset > 0)
2400:       TextRect.Top += Offset/2 + SideBuffer;
2401:    else
2402:       TextRect.Top += 2 + SideBuffer;
2403:
2404:    TextRect.Top += SideBuffer;
2405:
2406:    // Draw any menu item icons
2407:    if(Menu->Images != NULL)
2408:    {
2409:       Icon = new TIcon();
2410:       Menu->Images->GetIcon(MenuItem->ImageIndex,Icon);
2411:       ACanvas->Draw(5,ARect.Top+ItemOffset+1,Icon);
2412:       delete Icon;
2413:    }
2414:
2415:    // Draw the text
2416:    DrawText(ACanvas->Handle,Text.c_str(), TextLength, &TextRect, 0);
2417: }
2418: //---------------------------------------------------------------------------
2419:
2420: //---------------------------------------------------------------------------
2421:
2422: void __fastcall TfrmMainWindow::mnuFileClick(TObject *Sender)
2423: {
2424:    TMenuItem *MenuItem = ((TMenuItem*)Sender);
2425:    if(MenuItem->Count > 0)
2426:    {
2427:       for(int i=0; i <= MenuItem->Count-1; i++)
2428:       {
2429:          MenuItem->Items[i]->OnMeasureItem = MyExpandItemWidth;
2430:          MenuItem->Items[i]->OnDrawItem = MyDrawItem;
2431:          if(MenuItem->Items[i]->Count > 0)
2432:          {
2433:             for(int x=0; x <= MenuItem->Items[i]->Count-1; x++)
2434:             {
2435:                MenuItem->Items[i]->Items[x]->OnMeasureItem = MyExpandItemWidth;
2436:                MenuItem->Items[i]->Items[x]->OnDrawItem = MyDrawItem;
2437:             }
2438:          }
```

```
2439:       }
2440:    }
2441: }
2442:
2443: //---------------------------------------------------------------------------
2444:
2445: void __fastcall TfrmMainWindow::mnuManagementClick(TObject *Sender)
2446: {
2447:    TMenuItem *MenuItem = ((TMenuItem*)Sender);
2448:    if(MenuItem->Count > 0)
2449:    {
2450:       for(int i=0; i <= MenuItem->Count-1; i++)
2451:       {
2452:          MenuItem->Items[i]->OnMeasureItem = MyExpandItemWidth;
2453:          MenuItem->Items[i]->OnDrawItem = MyDrawItem;
2454:          if(MenuItem->Items[i]->Count > 0)
2455:          {
2456:             for(int x=0; x <= MenuItem->Items[i]->Count-1; x++)
2457:             {
2458:                MenuItem->Items[i]->Items[x]->OnMeasureItem = MyExpandItemWidth;
2459:                MenuItem->Items[i]->Items[x]->OnDrawItem = MyDrawItem;
2460:             }
2461:          }
2462:       }
2463:    }
2464: }
2465: //---------------------------------------------------------------------------
2466:
2467: void __fastcall TfrmMainWindow::mnuCustomerManagementClick(
2468:       TObject *Sender)
2469: {
2470:    TMenuItem *MenuItem = ((TMenuItem*)Sender);
2471:    if(MenuItem->Count > 0)
2472:    {
2473:       for(int i=0; i <= MenuItem->Count-1; i++)
2474:       {
2475:          MenuItem->Items[i]->OnMeasureItem = MyExpandItemWidth;
2476:          MenuItem->Items[i]->OnDrawItem = MyDrawItem;
2477:          if(MenuItem->Items[i]->Count > 0)
2478:          {
2479:             for(int x=0; x <= MenuItem->Items[i]->Count-1; x++)
2480:             {
2481:                MenuItem->Items[i]->Items[x]->OnMeasureItem = MyExpandItemWidth;
2482:                MenuItem->Items[i]->Items[x]->OnDrawItem = MyDrawItem;
2483:             }
2484:          }
```

```
2485:       }
2486:    }
2487: }
2488: //---------------------------------------------------------------------------
2489:
2490: void __fastcall TfrmMainWindow::mnuServicesClick(TObject *Sender)
2491: {
2492:    TMenuItem *MenuItem = ((TMenuItem*)Sender);
2493:    if(MenuItem->Count > 0)
2494:    {
2495:       for(int i=0; i <= MenuItem->Count-1; i++)
2496:       {
2497:          MenuItem->Items[i]->OnMeasureItem = MyExpandItemWidth;
2498:          MenuItem->Items[i]->OnDrawItem = MyDrawItem;
2499:          if(MenuItem->Items[i]->Count > 0)
2500:          {
2501:             for(int x=0; x <= MenuItem->Items[i]->Count-1; x++)
2502:             {
2503:                MenuItem->Items[i]->Items[x]->OnMeasureItem = MyExpandItemWidth;
2504:                MenuItem->Items[i]->Items[x]->OnDrawItem = MyDrawItem;
2505:             }
2506:          }
2507:       }
2508:    }
2509: }
2510: //---------------------------------------------------------------------------
2511:
2512: void __fastcall TfrmMainWindow::mnuHelpClick(TObject *Sender)
2513: {
2514:    TMenuItem *MenuItem = ((TMenuItem*)Sender);
2515:    if(MenuItem->Count > 0)
2516:    {
2517:       for(int i=0; i <= MenuItem->Count-1; i++)
2518:       {
2519:          MenuItem->Items[i]->OnMeasureItem = MyExpandItemWidth;
2520:          MenuItem->Items[i]->OnDrawItem = MyDrawItem;
2521:          if(MenuItem->Items[i]->Count > 0)
2522:          {
2523:             for(int x=0; x <= MenuItem->Items[i]->Count-1; x++)
2524:             {
2525:                MenuItem->Items[i]->Items[x]->OnMeasureItem = MyExpandItemWidth;
2526:                MenuItem->Items[i]->Items[x]->OnDrawItem = MyDrawItem;
2527:             }
2528:          }
2529:       }
2530:    }
```

```
2531: }
2532: //---------------------------------------------------------------------------
2533:
2534: void __fastcall TfrmMainWindow::fileLogoffClick(TObject *Sender)
2535: {
2536:    // ------------------------------------------------------------------
2537:    // Diable Menu System
2538:    // ------------------------------------------------------------------
2539:    mnuManagement->Enabled = false;
2540:    mnuCustomerManagement->Enabled = false;
2541:    mnuServices->Enabled = false;
2542:
2543:    fileLogoff->Enabled = false;
2544:    fileLogin->Enabled = true;
2545:    // ------------------------------------------------------------------
2546:
2547:    // Custom Outlook Like Bar Design
2548:    dcLeftPanel->ActiveGroupIndex = 0;
2549:
2550:    // Disable Left Panel
2551:    dcLeftPanel->Groups[0]->Enabled = false;
2552:    dcLeftPanel->Groups[1]->Enabled = false;
2553:    dcLeftPanel->Groups[2]->Enabled = false;
2554:    dcLeftPanel->Groups[3]->Enabled = false;
2555:
2556:    // Automatically Time Stamp the time card
2557:    dmHMS->MWstampTimeCard( ssn, startTime );
2558: }
2559: //---------------------------------------------------------------------------
2560:
2561: //---------------------------------------------------------------------------
2562: // Functions to handle Window Handle Creation
2563: //---------------------------------------------------------------------------
2564: void __fastcall TfrmMainWindow::createReservationWindow( )
2565: {
2566:    // Create the reservation window
2567:    reservation = new TreservationWindow( this );
2568:
2569:    reservation->Visible = true;
2570:    // reservation->txtReservationID->Enabled = false;
2571:    reservation->SetFocus( );
2572: }
2573:
2574: void __fastcall TfrmMainWindow::createLoginWindow( )
2575: {
2576:    // Create the login window
```

```
2577:    login = new TloginWindow( this );
2578:
2579:    login->Visible = true;
2580:    login->txtUserName->SetFocus( );
2581:    login->SetFocus( );
2582: }
2583:
2584: void __fastcall TfrmMainWindow::createAccountWindow( )
2585: {
2586:    // Create the account window
2587:    account = new TaccountWindow( this );
2588:
2589:    account->Visible = true;
2590:    account->SetFocus( );
2591:
2592:    if( !accountID.IsEmpty( ) )
2593:    {
2594:       account->txtAccountID->Text = customerID;
2595:       account->butAccountNumberClick( this );
2596:    }
2597: }
2598:
2599: void __fastcall TfrmMainWindow::createCustomerWindow( )
2600: {
2601:    // Create the customer window
2602:    customer = new TcustomerWindow( this );
2603:
2604:    customer->Visible = true;
2605:    customer->SetFocus( );
2606:
2607:    if( !customerID.IsEmpty( ) )
2608:    {
2609:       customer->txtCustomerID->Text = customerID;
2610:       customer->butCustomerIDClick( this );
2611:    }
2612: }
2613:
2614: void __fastcall TfrmMainWindow::createTransactionWindow( )
2615: {
2616:    // Create the transaction window
2617:    transaction = new TtransactionWindow( this );
2618:
2619:    transaction->Visible = true;
2620:    transaction->SetFocus( );
2621:
2622:    transaction->dtDate->Date = Date( );
```

```
2623:    transaction->txtTime->Text = TimeToStr( Time( ) );
2624: }
2625:
2626: void __fastcall TfrmMainWindow::createRoomTypeWindow( )
2627: {
2628:    // Create the room type window
2629:    roomType = new TroomTypeWindow( this );
2630:
2631:    roomType->Visible = true;
2632:    roomType->SetFocus( );
2633: }
2634:
2635: void __fastcall TfrmMainWindow::createEmployeeInformationWindow( )
2636: {
2637:    // Create the employee information window
2638:    employeeInformation = new TemployeeInformation( this );
2639:
2640:    employeeInformation->Visible = true;
2641:    employeeInformation->SetFocus( );
2642: }
2643:
2644: void __fastcall TfrmMainWindow::createEmployeeScheduleWindow( )
2645: {
2646:    // Create the employee schedule window
2647:    employeeSchedule = new TemployeeSchedule( this );
2648:
2649:    employeeSchedule->Visible = true;
2650:    employeeSchedule->SetFocus( );
2651: }
2652:
2653: void __fastcall TfrmMainWindow::createTimeCardWindow( )
2654: {
2655:    // Create the time card window
2656:    timeCard = new TtimeCard( this );
2657:
2658:    timeCard->Visible = true;
2659:    timeCard->SetFocus( );
2660: }
2661:
2662: void __fastcall TfrmMainWindow::createFoodServiceWindow( )
2663: {
2664:    // Create the food service window
2665:    foodService = new TfoodService( this );
2666:
2667:    foodService->Visible = true;
2668:    foodService->SetFocus( );
```

```
2669: }
2670:
2671: void __fastcall TfrmMainWindow::createRoomInformationWindow( )
2672: {
2673:    // Create the room information window
2674:    roomInformation = new TroomInformation( this );
2675:
2676:    roomInformation->Visible = true;
2677:    roomInformation->SetFocus( );
2678: }
2679: //---------------------------------------------------------------------------
2680:
2681: //---------------------------------------------------------------------------
2682: // Functions to handle Window Handle Destruction
2683: //---------------------------------------------------------------------------
2684: void __fastcall TfrmMainWindow::destroyReservationWindow( )
2685: {
2686:    reservation->Close( );
2687: }
2688:
2689: void __fastcall TfrmMainWindow::destroyLoginWindow( )
2690: {
2691:    login->Close( );
2692: }
2693:
2694: void __fastcall TfrmMainWindow::destroyAccountWindow( )
2695: {
2696:    account->Close( );
2697: }
2698:
2699: void __fastcall TfrmMainWindow::destroyCustomerWindow( )
2700: {
2701:    customer->Close( );
2702: }
2703:
2704: void __fastcall TfrmMainWindow::destroyTransactionWindow( )
2705: {
2706:    transaction->Close( );
2707: }
2708:
2709: void __fastcall TfrmMainWindow::destroyRoomTypeWindow( )
2710: {
2711:    roomType->Close( );
2712: }
2713:
2714: void __fastcall TfrmMainWindow::destroyEmployeeInformationWindow( )
```

```
2715: {
2716:    employeeInformation->Close( );
2717: }
2718:
2719: void __fastcall TfrmMainWindow::destroyEmployeeScheduleWindow( )
2720: {
2721:    employeeSchedule->Close( );
2722: }
2723:
2724: void __fastcall TfrmMainWindow::destroyTimeCardWindow( )
2725: {
2726:    timeCard->Close( );
2727: }
2728:
2729: void __fastcall TfrmMainWindow::destroyFoodServiceWindow( )
2730: {
2731:    foodService->Close( );
2732: }
2733:
2734: void __fastcall TfrmMainWindow::destroyRoomInformationWindow( )
2735: {
2736:    roomInformation->Close( );
2737: }
2738: //---------------------------------------------------------------------------
2739:
2740: void __fastcall TfrmMainWindow::helpContentClick(TObject *Sender)
2741: {
2742:    Application->HelpCommand( HELP_CONTENTS, 0 );
2743: }
2744: //---------------------------------------------------------------------------
2745:
2746:
2747: void __fastcall TfrmMainWindow::EmployeeInformation1Click(TObject *Sender)
2748: {
2749:    // Create Employee Information Window Handle
2750:    createEmployeeInformationWindow( );
2751: }
2752: //---------------------------------------------------------------------------
2753:
2754: void __fastcall TfrmMainWindow::EmployeeSchedule1Click(TObject *Sender)
2755: {
2756:    // Create Employee Schedule Window Handle
2757:    createEmployeeScheduleWindow( );
2758: }
2759: //---------------------------------------------------------------------------
2760:
```

```
2761: void __fastcall TfrmMainWindow::EmployeeTimeCard1Click(TObject *Sender)
2762: {
2763:    // Create Time Card Window Handle
2764:    createTimeCardWindow( );
2765: }
2766: //---------------------------------------------------------------------------
2767:
2768: void __fastcall TfrmMainWindow::servicesFoodClick(TObject *Sender)
2769: {
2770:    // Create Food Service Window Handle
2771:    createFoodServiceWindow( );
2772: }
2773: //---------------------------------------------------------------------------
2774:
2775: void __fastcall TfrmMainWindow::servicesRoomClick(TObject *Sender)
2776: {
2777:    // Create Room Information Window Handle
2778:    createRoomInformationWindow( );
2779: }
2780: //---------------------------------------------------------------------------
2781:
2782:
2783: void __fastcall TfrmMainWindow::dcLeftPanelButtonClick(TObject *Sender,
2784:       TListItem *Item)
2785: {
2786:    switch( dcLeftPanel->ActiveGroupIndex )
2787:    {
2788:       case 0:  // Reservation Panel
2789:       {
2790:          switch( Item->Index )
2791:          {
2792:             case 0:
2793:             {
2794:                if( reservation )
2795:                {
2796:                   reservation->Visible = true;
2797:                   reservation->SetFocus( );
2798:                }
2799:                else
2800:                {
2801:                   createReservationWindow( );
2802:                }
2803:                break;
2804:             }
2805:             case 1:
2806:             {
```

```
2807:                if( reservation )
2808:                {
2809:                   reservation->Visible = true;
2810:                   reservation->SetFocus( );
2811:                }
2812:                else
2813:                {
2814:                   createReservationWindow( );
2815:                }
2816:                break;
2817:             }
2818:             case 2:
2819:             {
2820:                if( reservation )
2821:                {
2822:                   reservation->Visible = true;
2823:                   reservation->SetFocus( );
2824:                }
2825:                else
2826:                {
2827:                   createReservationWindow( );
2828:                }
2829:                break;
2830:             }
2831:          }
2832:          break;
2833:       }
2834:
2835:       case 1:  // Services Panel
2836:       {
2837:          switch( Item->Index )
2838:          {
2839:             case 0:
2840:             {
2841:                if( foodService )
2842:                {
2843:                   foodService->Visible = true;
2844:                   foodService->SetFocus( );
2845:                }
2846:                else
2847:                {
2848:                   createFoodServiceWindow( );
2849:                }
2850:                break;
2851:             }
2852:             case 1:
```

```
2853:             {
2854:                // code for Movie services
2855:                break;
2856:             }
2857:             case 2:
2858:             {
2859:                if( roomInformation )
2860:                {
2861:                   roomInformation->Visible = true;
2862:                   roomInformation->SetFocus( );
2863:                }
2864:                else
2865:                {
2866:                   createRoomInformationWindow( );
2867:                }
2868:                break;
2869:             }
2870:          }
2871:          break;
2872:       }
2873:
2874:       case 2:  // Management Panel
2875:       {
2876:          switch( Item->Index )
2877:          {
2878:             case 0:
2879:             {
2880:                if( employeeInformation )
2881:                {
2882:                   employeeInformation->Visible = true;
2883:                   employeeInformation->SetFocus( );
2884:                }
2885:                else
2886:                {
2887:                   createEmployeeInformationWindow( );
2888:                }
2889:                break;
2890:             }
2891:             case 1:
2892:             {
2893:                if( employeeSchedule )
2894:                {
2895:                   employeeSchedule->Visible = true;
2896:                   employeeSchedule->SetFocus( );
2897:                }
2898:                else
```

```
2899:                {
2900:                   createEmployeeScheduleWindow( );
2901:                }
2902:                break;
2903:             }
2904:             case 2:
2905:             {
2906:                if( timeCard )
2907:                {
2908:                   timeCard->Visible = true;
2909:                   timeCard->SetFocus( );
2910:                }
2911:                else
2912:                {
2913:                   createTimeCardWindow( );
2914:                }
2915:                break;
2916:             }
2917:          }
2918:          break;
2919:       }
2920:
2921:       case 3:  // Employee Panel
2922:       {
2923:          switch( Item->Index )
2924:          {
2925:             case 0:
2926:             {
2927:                if( timeCard )
2928:                {
2929:                   timeCard->Visible = true;
2930:                   timeCard->SetFocus( );
2931:                }
2932:                else
2933:                {
2934:                   createTimeCardWindow( );
2935:                }
2936:                break;
2937:             }
2938:          }
2939:          break;
2940:       }
2941:    }
2942: }
2943: //---------------------------------------------------------------------------
2944:
```

```
2945:
2946:
2947:
2948: //---------------------------------------------------------------------------
2949:
2950: #include <vcl.h>
2951: #pragma hdrstop
2952: USERES("HMS.res");
2953: USEFORM("mainWindow.cpp", frmMainWindow);
2954: USEFORM("frmLogin.cpp", loginWindow);
2955: USEFORM("frmReservation.cpp", reservationWindow);
2956: USEFORM("frmCustomer.cpp", customerWindow);
2957: USEFORM("frmAccount.cpp", accountWindow);
2958: USEFORM("dataModule.cpp", dmHMS); /* TDataModule: File Type */
2959: USEFORM("transactoinWindow.cpp", transactionWindow);
2960: USEFORM("frmRoomType.cpp", roomTypeWindow);
2961: USEFORM("frmEmployeeInformation.cpp", employeeInformation);
2962: USEFORM("frmEmployeeSchedule.cpp", employeeSchedule);
2963: USEFORM("frmTimeCard.cpp", timeCard);
2964: USEFORM("frmFoodService.cpp", foodService);
2965: USEFORM("frmRoomInformation.cpp", roomInformation);
2966: //---------------------------------------------------------------------------
2967: WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
2968: {
2969:    try
2970:    {
2971:       Application->Initialize();
2972:       Application->CreateForm(__classid(TfrmMainWindow), &frmMainWindow);
2973:        Application->CreateForm(__classid(TdmHMS), &dmHMS);
2974:        Application->Run();
2975:    }
2976:    catch (Exception &exception)
2977:    {
2978:       Application->ShowException(&exception);
2979:    }
2980:    return 0;
2981: }
2982: //---------------------------------------------------------------------------
2983:
2984:
2985:
2986: Copyright - Vahe Karamian - www.karamian.com 2001r.
```