

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Text.RegularExpressions;
11 using MySql.Data.MySqlClient;
12 using System.Data.SqlClient;
13 using System.IO;
14
15 namespace customerSignUp
16 {
17     public partial class Billing : Form
18     {
19         private string guest0firstName="", guest0lastName="", guest0address="",
20             guest0city="", guest0country="", guest0email="", numberOfGuest="",
21             tripId="", guest0phone="", shipName="", departureDate="",
22             departureDateSql="", tripDetails="", deckName="",
23             roomNumber="",roomType="";
24
25         private DateTime unselectedDateBirthGus1, unselectedDateBirthGus2,
26             unselectedDateBirthGus3, dateOfBirthGus0, departureDateForCalculate,now
27             = DateTime.Today;
28
29         private passValuesToForm cs = new passValuesToForm();
30         private MySqlDataReader sqlReadCity;
31         private int num0fGuest,roomTypeId;
32
33         /////////////////////////////////// Passing of values from previous
34         forms ///////////////////////////////////
35
36         public Billing(string fName, string lName, string ph, string addres,
37             string num0fGuest, string shpName, string eml, string county, string
38             cty, DateTime dobGuest0, string departDate, string
39             departDateSql,DateTime departDateForCal, string tripid,string
40             tripInfo,string deckNam,string roomNum,int roomTypeId)
41         {
42             InitializeComponent();
43
44             guest0firstName = cs.passFnameValue = fName;
45             guest0lastName = cs.passLnameValue = lName;
46             guest0phone = cs.passPhoneValue = ph;
47             guest0address = cs.passAddressValue = addres;
48             guest0country = cs.passCountryValue = county;
49             guest0email = cs.passEmailValue = eml;
50             shipName = cs.passShipNameValue = shpName;
51             numberOfGuest = cs.passNumberOfGuestValue = num0fGuest;
52             guest0city = cs.passCityValue = cty;
53             dateOfBirthGus0 = cs.passDateOfBirthGus0Value = dobGuest0;
54             departureDate = cs.passDepartureDateValue = departDate;

```

```

42      departureDateSql = cs.passDepartureDateSqlValue = departDateSql;
43      departureDateForCalculate = cs.passDepartureDateForCal =
      departDateForCal;
44      tripId = cs.passTripIdValue = tripid;
45      tripDetails = cs.passTripDetailsValue = tripInfo;
46      deckName = deckNam;
47      roomNumber = roomNum;
48      roomTypeId = roomTypId;
49  }
50  private void Billing_Load(object sender, EventArgs e)
51  {
52
53      dateBirthGuest1Input.Format = DateTimePickerFormat.Custom;
54      dateBirthGuest1Input.CustomFormat = "MMMM dd, yyyy";
55      unselectedDateBirthGus1 = dateBirthGuest1Input.Value;
56
57      dateBirthGuest2Input.Format = DateTimePickerFormat.Custom;
58      dateBirthGuest2Input.CustomFormat = "MMMM dd, yyyy";
59      unselectedDateBirthGus2 = dateBirthGuest2Input.Value;
60
61      dateBirthGuest3Input.Format = DateTimePickerFormat.Custom;
62      dateBirthGuest3Input.CustomFormat = "MMMM dd, yyyy";
63      unselectedDateBirthGus3 = dateBirthGuest3Input.Value;
64
65      numOfGuest = int.Parse(numberOfGuest);
66
67      //////////// Adjust UX design accordingly to the number of
      guest selected on Main form  ////////////
68      switch (numOfGuest)
69      {
70          case 0:
71              this.Size = new Size(1052, 445);
72              labelInvoice.Location = new Point(445, 0);
73              displayRes.Location = new Point(111, 87);
74              displayRes.Size = new Size(638, 247);
75              btnExit.Location = new Point(811, 217);
76              btnCalc.Location = new Point(811, 116);
77
78              labelGuest1.Dispose();
79              horizontalGuest1.Dispose();
80              labelFnameGuest1.Dispose();
81              labelLnameGuest1.Dispose();
82              labelDateBirthGuest1.Dispose();
83              fNameGuest1Input.Dispose();
84              lNameGuest1Input.Dispose();
85              dateBirthGuest1Input.Dispose();
86              validationDateBirthGuest1.Dispose();
87              validationFnameGuest1.Dispose();
88              validationLnameGuest1.Dispose();
89
90              labelGuest2.Dispose();
91              horizontalGuest2.Dispose();

```

```
92         labelFnameGuest2.Dispose();
93         labelLnameGuest2.Dispose();
94         labelDateBirthGuest2.Dispose();
95         fNameGuest2Input.Dispose();
96         lNameGuest2Input.Dispose();
97         dateBirthGuest2Input.Dispose();
98         validationDateBirthGuest2.Dispose();
99         validationFnameGuest2.Dispose();
100        validationLnameGuest2.Dispose();
101
102        labelGuest3.Dispose();
103        horizontalGuest3.Dispose();
104        labelFnameGuest3.Dispose();
105        labelLnameGuest3.Dispose();
106        labelDateBirthGuest3.Dispose();
107        fNameGuest3Input.Dispose();
108        lNameGuest3Input.Dispose();
109        dateBirthGuest3Input.Dispose();
110        validationDateBirthGuest3.Dispose();
111        validationFnameGuest3.Dispose();
112        validationLnameGuest3.Dispose();
113        break;
114
115        case 1:
116            this.Size = new Size(1052, 578);
117            labelInvoice.Location = new Point(445, 122);
118            displayRes.Location = new Point(111, 204);
119            displayRes.Size = new Size(638, 285);
120            btnExit.Location = new Point(811, 352);
121            btnCalc.Location = new Point(811, 251);
122
123            labelGuest2.Dispose();
124            horizontalGuest2.Dispose();
125            labelFnameGuest2.Dispose();
126            labelLnameGuest2.Dispose();
127            labelDateBirthGuest2.Dispose();
128            fNameGuest2Input.Dispose();
129            lNameGuest2Input.Dispose();
130            dateBirthGuest2Input.Dispose();
131            validationDateBirthGuest2.Dispose();
132            validationFnameGuest2.Dispose();
133            validationLnameGuest2.Dispose();
134
135            labelGuest3.Dispose();
136            horizontalGuest3.Dispose();
137            labelFnameGuest3.Dispose();
138            labelLnameGuest3.Dispose();
139            labelDateBirthGuest3.Dispose();
140            fNameGuest3Input.Dispose();
141            lNameGuest3Input.Dispose();
142            dateBirthGuest3Input.Dispose();
143            validationDateBirthGuest3.Dispose();
```

```
144         validationFnameGuest3.Dispose();
145         validationLnameGuest3.Dispose();
146         break;
147
148     case 2:
149         this.Size = new Size(1052, 710);
150         labelInvoice.Location = new Point(445, 241);
151         displayRes.Location = new Point(111, 331);
152         displayRes.Size = new Size(638, 285);
153         btnExit.Location = new Point(811, 471);
154         btnCalc.Location = new Point(811, 370);
155
156         labelGuest3.Dispose();
157         horizontalGuest3.Dispose();
158         labelFnameGuest3.Dispose();
159         labelLnameGuest3.Dispose();
160         labelDateBirthGuest3.Dispose();
161         fNameGuest3Input.Dispose();
162         lNameGuest3Input.Dispose();
163         dateBirthGuest3Input.Dispose();
164         validationDateBirthGuest3.Dispose();
165         validationFnameGuest3.Dispose();
166         validationLnameGuest3.Dispose();
167         break;
168
169     case 3:
170         break;
171 }
172
173 }
174 private void btnCalc_Click_1(object sender, EventArgs e)
175 {
176     bool wrong = false;
177
178     //Validation of names and dob for Guest 1 to 3
179     switch (numOfGuest)
180     {
181     case 0:
182         break;
183     case 1:
184         if (inputValidation.checkNames(fNameGuest1Input.Text.ToString()
185             ()) && fNameGuest1Input.Text.Length > 2)
186         {
187             validationFnameGuest1.ForeColor = Color.Black;
188             validationFnameGuest1.Text = string.Empty;
189         }
190         else
191         {
192             validationFnameGuest1.ForeColor = Color.Red;
193             validationFnameGuest1.Text = "x";
194             wrong = true;
195         }
196     }
197 }
```

```
194     }
195
196     if (inputValidation.checkNames(lNameGuest1Input.Text.ToString ➤
197     ()) && lNameGuest1Input.Text.Length > 1)
198     {
199         validationLnameGuest1.ForeColor = Color.Black;
200         validationLnameGuest1.Text = string.Empty;
201     }
202     else
203     {
204         validationLnameGuest1.ForeColor = Color.Red;
205         validationLnameGuest1.Text = "x";
206         wrong = true;
207     }
208
209     if (dateBirthGuest1Input.Value != unselectedDateBirthGus1)
210     {
211         validationDateBirthGuest1.ForeColor = Color.Black;
212         validationDateBirthGuest1.Text = string.Empty;
213     }
214     else
215     {
216         validationDateBirthGuest1.ForeColor = Color.Red;
217         validationDateBirthGuest1.Text = "x";
218         wrong = true;
219     }
220     break;
221 case 2:
222     //////////////// validate guest 1 ////////////////
223     if (inputValidation.checkNames(fNameGuest1Input.Text.ToString ➤
224     ()) && fNameGuest1Input.Text.Length > 2)
225     {
226         validationFnameGuest1.ForeColor = Color.Black;
227         validationFnameGuest1.Text = string.Empty;
228     }
229     else
230     {
231         validationFnameGuest1.ForeColor = Color.Red;
232         validationFnameGuest1.Text = "x";
233         wrong = true;
234     }
235
236     if (inputValidation.checkNames(lNameGuest1Input.Text.ToString ➤
237     ()) && lNameGuest1Input.Text.Length > 1)
238     {
239         validationLnameGuest1.ForeColor = Color.Black;
240         validationLnameGuest1.Text = string.Empty;
241     }
242     else
243     {
244         validationLnameGuest1.ForeColor = Color.Red;
245         validationLnameGuest1.Text = "x";
```

```
243         wrong = true;
244     }
245
246     if (dateBirthGuest1Input.Value != unselectedDateBirthGus1)
247     {
248         validationDateBirthGuest1.ForeColor = Color.Black;
249         validationDateBirthGuest1.Text = string.Empty;
250     }
251     else
252     {
253         validationDateBirthGuest1.ForeColor = Color.Red;
254         validationDateBirthGuest1.Text = "x";
255         wrong = true;
256     }
257
258
259     //////////// guest 2 ////////////
260     if (inputValidation.checkNames(fNameGuest2Input.Text.ToString
261     ()) && fNameGuest2Input.Text.Length > 2)
262     {
263         validationFnameGuest2.ForeColor = Color.Black;
264         validationFnameGuest2.Text = string.Empty;
265     }
266     else
267     {
268         validationFnameGuest2.ForeColor = Color.Red;
269         validationFnameGuest2.Text = "x";
270         wrong = true;
271     }
272
273     if (inputValidation.checkNames(lNameGuest2Input.Text.ToString
274     ()) && lNameGuest2Input.Text.Length > 1)
275     {
276         validationLnameGuest2.ForeColor = Color.Black;
277         validationLnameGuest2.Text = string.Empty;
278     }
279     else
280     {
281         validationLnameGuest2.ForeColor = Color.Red;
282         validationLnameGuest2.Text = "x";
283         wrong = true;
284     }
285
286     if (dateBirthGuest2Input.Value != unselectedDateBirthGus2)
287     {
288         validationDateBirthGuest2.ForeColor = Color.Black;
289         validationDateBirthGuest2.Text = string.Empty;
290     }
291     else
292     {
293         validationDateBirthGuest2.ForeColor = Color.Red;
294         validationDateBirthGuest2.Text = "x";
295     }
```

```
293         wrong = true;
294     }
295     break;
296 case 3:
297     //////////////// validate guest 1 ////////////////
298     if (inputValidation.checkNames(fNameGuest1Input.Text.ToString
299     ()) && fNameGuest1Input.Text.Length > 2)
300     {
301         validationFnameGuest1.ForeColor = Color.Black;
302         validationFnameGuest1.Text = string.Empty;
303     }
304     else
305     {
306         validationFnameGuest1.ForeColor = Color.Red;
307         validationFnameGuest1.Text = "x";
308         wrong = true;
309     }
310     if (inputValidation.checkNames(lNameGuest1Input.Text.ToString
311     ()) && lNameGuest1Input.Text.Length > 1)
312     {
313         validationLnameGuest1.ForeColor = Color.Black;
314         validationLnameGuest1.Text = string.Empty;
315     }
316     else
317     {
318         validationLnameGuest1.ForeColor = Color.Red;
319         validationLnameGuest1.Text = "x";
320         wrong = true;
321     }
322     if (dateBirthGuest1Input.Value != unselectedDateBirthGus1)
323     {
324         validationDateBirthGuest1.ForeColor = Color.Black;
325         validationDateBirthGuest1.Text = string.Empty;
326     }
327     else
328     {
329         validationDateBirthGuest1.ForeColor = Color.Red;
330         validationDateBirthGuest1.Text = "x";
331         wrong = true;
332     }
333
334     //////////////// guest 2 ////////////////
335     if (inputValidation.checkNames(fNameGuest2Input.Text.ToString
336     ()) && fNameGuest2Input.Text.Length > 2)
337     {
338         validationFnameGuest2.ForeColor = Color.Black;
339         validationFnameGuest2.Text = string.Empty;
340     }
341     else
342     {
```

```
342         validationFnameGuest2.ForeColor = Color.Red;
343         validationFnameGuest2.Text = "x";
344         wrong = true;
345     }
346
347     if (inputValidation.checkNames(lNameGuest2Input.Text.ToString
348     ()) && lNameGuest2Input.Text.Length > 1)
349     {
350         validationLnameGuest2.ForeColor = Color.Black;
351         validationLnameGuest2.Text = string.Empty;
352     }
353     else
354     {
355         validationLnameGuest2.ForeColor = Color.Red;
356         validationLnameGuest2.Text = "x";
357         wrong = true;
358     }
359
360     if (dateBirthGuest2Input.Value != unselectedDateBirthGus2)
361     {
362         validationDateBirthGuest2.ForeColor = Color.Black;
363         validationDateBirthGuest2.Text = string.Empty;
364     }
365     else
366     {
367         validationDateBirthGuest2.ForeColor = Color.Red;
368         validationDateBirthGuest2.Text = "x";
369         wrong = true;
370     }
371
372     //////////////////////////////////// Guest
373     3  ////////////////////////////////////
374
375     if (inputValidation.checkNames(fNameGuest3Input.Text.ToString
376     ()) && fNameGuest3Input.Text.Length > 2)
377     {
378         validationFnameGuest3.ForeColor = Color.Black;
379         validationFnameGuest3.Text = string.Empty;
380     }
381     else
382     {
383         validationFnameGuest3.ForeColor = Color.Red;
384         validationFnameGuest3.Text = "x";
385         wrong = true;
386     }
387
388     if (inputValidation.checkNames(lNameGuest3Input.Text.ToString
389     ()) && lNameGuest3Input.Text.Length > 1)
390     {
391         validationLnameGuest3.ForeColor = Color.Black;
392         validationLnameGuest3.Text = string.Empty;
393     }
394     else
395     {
396         validationLnameGuest3.ForeColor = Color.Red;
397         validationLnameGuest3.Text = "x";
398         wrong = true;
399     }
```



```
390         validationLnameGuest3.ForeColor = Color.Red;
391         validationLnameGuest3.Text = "x";
392         wrong = true;
393     }
394
395     if (dateBirthGuest3Input.Value != unselectedDateBirthGus3)
396     {
397         validationDateBirthGuest3.ForeColor = Color.Black;
398         validationDateBirthGuest3.Text = string.Empty;
399     }
400     else
401     {
402         validationDateBirthGuest3.ForeColor = Color.Red;
403         validationDateBirthGuest3.Text = "x";
404         wrong = true;
405     }
406     break;
407 }
408
409 if (wrong == false)
410 {
411
412     string guest0name = null, guest1name, guest2name, guest3name,
413     totalGuestNames = null;
414     int senior = 65, numOfSenior = 0, numOfAdults = 0;
415     DateTime now = DateTime.Today;
416     int ageDiffGuest0, ageDiffGuest1, ageDiffGuest2, ageDiffGuest3;
417
418     displayRes.Text = "";
419
420     //////////////////////////////////// Calculate ages for all
421     guests ////////////////////////////////////
422     switch (numOfGuest)
423     {
424     case 0:
425         ageDiffGuest0 = now.Year - dateOfBirthGus0.Year;
426         if (ageDiffGuest0 >= senior)
427         {
428             guest0name = guest0firstNam + " " + guest0lastNam +
429             " (Senior)";
430             numOfSenior += 1;
431         }
432         else
433         {
434             guest0name = guest0firstNam + " " + guest0lastNam +
435             " (Adult)";
436             numOfAdults += 1;
437         }
438     }
439     break;
440 case 1:
441     ageDiffGuest0 = now.Year - dateOfBirthGus0.Year;
```

```
438         if (ageDiffGuest0 >= senior)
439         {
440             guest0name = guest0firstNam + " " + guest0lastNam +
441             " (Senior)";
442             numOfSenior += 1;
443         }
444         else
445         {
446             guest0name = guest0firstNam + " " + guest0lastNam +
447             " (Adult)";
448             numOfAdults += 1;
449             ageDiffGuest1 = now.Year -
450             dateBirthGuest1Input.Value.Year;
451             if (ageDiffGuest1 >= senior)
452             {
453                 guest1name = fNameGuest1Input.Text + " " +
454                 lNameGuest1Input.Text + " (Senior)";
455                 totalGuestNames += "Guest 1: " + guest1name +
456                 "\n";
457                 numOfSenior += 1;
458             }
459             else
460             {
461                 guest1name = fNameGuest1Input.Text + " " +
462                 lNameGuest1Input.Text + " (Adult)";
463                 totalGuestNames += "Guest 1: " + guest1name +
464                 "\n";
465                 numOfAdults += 1;
466             }
467             break;
468         case 2:
469             ageDiffGuest0 = now.Year - dateOfBirthGus0.Year;
470             if (ageDiffGuest0 >= senior)
471             {
472                 guest0name = guest0firstNam + " " + guest0lastNam +
473                 " (Senior)";
474                 numOfSenior += 1;
475             }
476             else
477             {
478                 guest0name = guest0firstNam + " " + guest0lastNam +
479                 " (Adult)";
480                 numOfAdults += 1;
481             }
482             ageDiffGuest1 = now.Year -
483             dateBirthGuest1Input.Value.Year;
484             if (ageDiffGuest1 >= senior)
485             {
486                 guest1name = fNameGuest1Input.Text + " " +
487                 lNameGuest1Input.Text + " (Senior)";
488                 totalGuestNames += "Guest 1: " + guest1name +
```

```

        "\n";
        numOfSenior += 1;
    }
    else
    {
        guest1name = fNameGuest1Input.Text + " " +
lNameGuest1Input.Text + " (Adult)";
        totalGuestNames += "Guest 1: " + guest1name +
        "\n";
        numOfAdults += 1;
    }

    ageDiffGuest2 = now.Year -
dateBirthGuest2Input.Value.Year;
    if (ageDiffGuest2 >= senior)
    {
        guest2name = fNameGuest2Input.Text + " " +
lNameGuest2Input.Text + " (Senior)";
        totalGuestNames += "Guest 2: " + guest2name +
        "\n";
        numOfSenior += 1;
    }
    else
    {
        guest2name = fNameGuest2Input.Text + " " +
lNameGuest2Input.Text + " (Adult)";
        totalGuestNames += "Guest 2: " + guest2name +
        "\n";
        numOfAdults += 1;
    }
    break;
case 3:
    ageDiffGuest0 = now.Year - dateOfBirthGus0.Year;
    if (ageDiffGuest0 >= senior)
    {
        guest0name = guest0firstNam + " " + guest0lastNam +
" (Senior)";
        numOfSenior += 1;
    }
    else
    {
        guest0name = guest0firstNam + " " + guest0lastNam +
" (Adult)";
        numOfAdults += 1;
    }
    ageDiffGuest1 = now.Year -
dateBirthGuest1Input.Value.Year;
    if (ageDiffGuest1 >= senior)
    {
        guest1name = fNameGuest1Input.Text + " " +
lNameGuest1Input.Text + " (Senior)";

```

```
519         totalGuestNames += "Guest 1: " + guest1name +  
        "\n";  
520         numOfSenior += 1;  
521     }  
522     else  
523     {  
524         guest1name = fNameGuest1Input.Text + " " +  
        lNameGuest1Input.Text + " (Adult)";  
525         totalGuestNames += "Guest 1: " + guest1name +  
        "\n";  
526         numOfAdults += 1;  
527     }  
528  
529  
530     ageDiffGuest2 = now.Year -  
        dateBirthGuest2Input.Value.Year;  
531     if (ageDiffGuest2 >= senior)  
532     {  
533         guest2name = fNameGuest2Input.Text + " " +  
        lNameGuest2Input.Text + " (Senior)";  
534         totalGuestNames += "Guest 2: " + guest2name +  
        "\n";  
535         numOfSenior += 1;  
536     }  
537     else  
538     {  
539         guest2name = fNameGuest2Input.Text + " " +  
        lNameGuest2Input.Text + " (Adult)";  
540         totalGuestNames += "Guest 2: " + guest2name +  
        "\n";  
541         numOfAdults += 1;  
542     }  
543  
544     ageDiffGuest3 = now.Year -  
        dateBirthGuest3Input.Value.Year;  
545     if (ageDiffGuest3 >= senior)  
546     {  
547         guest3name = fNameGuest3Input.Text + " " +  
        lNameGuest3Input.Text + " (Senior)";  
548         totalGuestNames += "Guest 3: " + guest3name +  
        "\n";  
549         numOfSenior += 1;  
550     }  
551     else  
552     {  
553         guest3name = fNameGuest3Input.Text + " " +  
        lNameGuest3Input.Text + " (Adult)";  
554         totalGuestNames += "Guest 3: " + guest3name +  
        "\n";  
555         numOfAdults += 1;  
556     }  
557     break;
```

```

558     }
559
560     ////////////////////////////////////////////////// room
561     prices ///////////////////////////////////
562     string totalTypeOfPass = "", displayPerPrice = "", displayTotal =
563     "", displayPerAdultCost = "", displayPerSeniorCost = "";
564     double seniorPrice = 0.00, totalCost = 0.00;
565     double roomPrice = 0.00, seniorRate = 0.80, closerToDepartRate =
566     0.60;
567
568     DBConnect cb = new DBConnect();
569     string query = "select * from ROOM_CATEGORY where roomCat_id =" +
570     roomTypeId + ";";
571     MySqlConnection conDataBase = new MySqlConnection
572     (cb.getConnection());
573     MySqlCommand cmdDataBase = new MySqlCommand(query, conDataBase);
574     MySqlDataReader myReader;
575
576     try
577     {
578         conDataBase.Open();
579         myReader = cmdDataBase.ExecuteReader();
580         while (myReader.Read()) // while READING
581         {
582             roomPrice = double.Parse(myReader
583             ["roomCat_price"].ToString());
584             roomType = myReader["roomCat_name"].ToString().Trim();
585         }
586
587         myReader.Close();
588     }
589     catch (Exception ex)
590     {
591         MessageBox.Show(ex.Message);
592     }
593     conDataBase.Close();
594
595     /////////////// Total price calculation ///////////////
596     seniorPrice = roomPrice * seniorRate;
597     double adultSpecialPrice = roomPrice * closerToDepartRate,
598     seniorSpecialPrice = seniorPrice * closerToDepartRate;
599     displayPerAdultCost = String.Format("{0:0.00}", roomPrice);
600     displayPerSeniorCost = String.Format("{0:0.00}", seniorPrice);
601     string specialDepartPriceDesp = "close to departure special\n",
602     seniorPriceDesp = "Senior price";
603     if (numOfAdults > 0)
604     {
605         totalTypeOfPass += numOfAdults + " Adult ";
606         if ((departureDateForCalculate - now).TotalDays > 20)
607         {
608             displayPerPrice += "1 x Adult is $" +

```

```

        displayPerAdultCost;
        totalCost += numOfAdults * roomPrice;
    }
    else
    {
        displayPerAdultCost = String.Format("{0:0.00}",
        adultSpecialPrice);
        displayPerPrice += "1 x Adult is $" + displayPerAdultCost
        + " (40% off) " + specialDepartPriceDesp;
        totalCost += numOfAdults * (roomPrice *
        closerToDepartRate);
    }
}
if (numOfSenior > 0)
{
    totalTypeOfPass += numOfSenior + " Senior ";
    if ((departureDateForCalculate - now).TotalDays > 20) //
    special depart price will allow if less than 20days
    {
        displayPerPrice += " 1 x Senior is $" +
        displayPerSeniorCost + " (20% off) " + seniorPriceDesp;
        totalCost += numOfSenior * seniorPrice;
    }
    else
    {
        displayPerSeniorCost = String.Format("{0:0.00}",
        seniorSpecialPrice);
        displayPerPrice += "1 x Senior is $" +
        displayPerSeniorCost + " (60% off) " +
        specialDepartPriceDesp;
        totalCost += numOfSenior * (seniorPrice *
        closerToDepartRate);
    }
}

displayTotal = String.Format("{0:0.00}", totalCost);

////////// Print
receipt ///////////////////////////////////
630 displayRes.Text = "Name: " + guest0name + "\n" +
    "Itinerary: " + tripDetails + "\n" + "Ship: ";
631 displayRes.Text += shipName + "\n" + "Depart: " +
    departureDate + "\n" + "Cabin: " + roomNumber + " (" +
    roomType + ") " + "\n";
632 displayRes.Text += "Deck Level: " + deckName + "\n\n" +
    "Passenger: " + totalTypeOfPass.Trim() + "\n";
633 displayRes.Text += totalGuestNames + "\n" + displayPerPrice.Trim
    () + "\n" + "Overall cost(tax included): $" + displayTotal;

634
635
636 string fileName = "lu_300234869";
637 using (StreamWriter writer = new StreamWriter(fileName + ".txt"))

```

```

638     {
639         writer.WriteLine("Name:      " + guest0name);
640         writer.WriteLine("Itinerary:  " + tripDetails);
641         writer.WriteLine("Ship:      "+ shipName);
642         writer.WriteLine("Depart:    " + departureDate);
643         writer.WriteLine("Cabin:      " + roomNumber + " (" +      ↗
        roomType + ") ");
644         writer.WriteLine("Deck Level:  " + deckName);
645         writer.WriteLine();
646         writer.WriteLine("Passenger:  " + totalTypeOfPass.Trim());
647         writer.WriteLine(totalGuestNames);
648         writer.WriteLine(displayPerPrice.Trim());
649         writer.WriteLine("Overall cost(tax included): $" +      ↗
        displayTotal);
650
651         writer.Close();
652     }
653     MessageBox.Show("File Saved");
654
655
656
657     //////////////// If user inputted city is found in db, do not      ↗
        insert. And otherwise..  //////////////////////////
658     DBConnect db = new DBConnect();
659
660     MySqlConnection con = new MySqlConnection(db.getConnection());
661     string citySelected = null;
662     con.Open();//to open sql Connection
663     MySqlCommand comm = new MySqlCommand();
664     comm.CommandText = "select city_name from CITY inner join COUNTRY      ↗
        on CITY.country_id = COUNTRY.country_id ";
665     comm.CommandText += "where COUNTRY.country_name = @countryName      ↗
        and CITY.city_name = @cityName ";
666     comm.Parameters.AddWithValue("@countryName", guest0country);
667     comm.Parameters.AddWithValue("@cityName", guest0city);
668
669     comm.Connection = con;
670     sqlReadCity = comm.ExecuteReader();
671
672     while (sqlReadCity.Read())
673     {
674         citySelected = (sqlReadCity["city_name"]).ToString();
675     }
676     sqlReadCity.Close();
677     con.Close();
678
679     DBConnect d = new DBConnect();
680     int incrementPhoneId = getLastPK("PHONE", "phone_id") + 1;
681     using (MySqlConnection conn = new MySqlConnection(d.getConnection      ↗
        ()))
682     {
683         using (MySqlCommand command = new MySqlCommand())

```

```

684         {
685             command.Connection = conn;
686             command.CommandType = CommandType.Text;
687             command.CommandText = "";
688             insertphone(conn, command, incrementPhoneId,
        guest0phone);
689             command.CommandText = "";
690
691             string cityNameSelected = "";
692             int incrementCityId = getLastPK("CITY", "city_id") + 1;
693             if (citySelected == null)
694             {
695                 insertCity(conn, command, incrementCityId,
        guest0country, guest0city);
696                 command.CommandText = "";
697                 cityNameSelected = guest0city;
698
699             }
700             else
701             {
702                 cityNameSelected = citySelected;
703             }
704
705
706             ////////////////////////////////// Insert Guests info into
        database //////////////////////////////////
707             int getLastPhoneId = getLastPK("PHONE", "phone_id");
708             int getLastAddId = getLastPK("ADDRESS", "address_id") +
        1;
709             int incrementAddressId = getLastPK("ADDRESS",
        "address_id") + 1;
710             int incrementPassengerId = getLastPK("PASSENGER",
        "pass_id") + 1;
711
712             string dateToSqlGuest0 = dateOfBirthGus0.ToString("yyyy-
        MM-dd").Trim();
713
714             DateTime dobGuest1 = dateBirthGuest1Input.Value;
715             string dateToSqlGuest1 = dobGuest1.ToString("yyyy-MM-
        dd").Trim();
716
717             DateTime dobGuest2 = dateBirthGuest2Input.Value;
718             string dateToSqlGuest2 = dobGuest2.ToString("yyyy-MM-
        dd").Trim();
719
720             DateTime dobGuest3 = dateBirthGuest3Input.Value;
721             string dateToSqlGuest3 = dobGuest3.ToString("yyyy-MM-
        dd").Trim();
722
723
724             switch (numOfGuest)
725             {

```



```
726         case 0://guest0 info
727             insertAddress(conn, command, incrementAddressId,
guest0address, cityNameSelected);
728             command.CommandText = "";
729             insertPassenger(conn, command,
incrementPassengerId, guest0firstNam, guest0lastNam,
getLastPhoneId, getLastAddId, guest0email, dateToSqlGuest0);
730             command.CommandText = "";
731             break;
732
733         case 1:
734             insertAddress(conn, command, incrementAddressId,
guest0address, cityNameSelected);
735             command.CommandText = "";
736             insertPassenger(conn, command,
incrementPassengerId, guest0firstNam, guest0lastNam,
getLastPhoneId, getLastAddId, guest0email, dateToSqlGuest0);
737             command.CommandText = "";
738
739             insertPassenger(conn, command,
incrementPassengerId + 1, fNameGuest1Input.Text,
lNameGuest1Input.Text, getLastPhoneId, getLastAddId,
guest0email, dateToSqlGuest1);
740             command.CommandText = "";
741             break;
742
743         case 2:
744             insertAddress(conn, command, incrementAddressId,
guest0address, cityNameSelected);
745             command.CommandText = "";
746             insertPassenger(conn, command,
incrementPassengerId, guest0firstNam, guest0lastNam,
getLastPhoneId, getLastAddId, guest0email, dateToSqlGuest0);
747
748             insertPassenger(conn, command,
incrementPassengerId + 1, fNameGuest1Input.Text,
lNameGuest1Input.Text, getLastPhoneId, getLastAddId,
guest0email, dateToSqlGuest1);
749             command.CommandText = "";
750             insertPassenger(conn, command,
incrementPassengerId + 2, fNameGuest2Input.Text,
lNameGuest2Input.Text, getLastPhoneId, getLastAddId,
guest0email, dateToSqlGuest2);
751             command.CommandText = "";
752             break;
753
754         case 3:
755             insertAddress(conn, command, incrementAddressId,
guest0address, cityNameSelected);
756             command.CommandText = "";
757             insertPassenger(conn, command,
incrementPassengerId, guest0firstNam, guest0lastNam,
```



```

797     MySqlCommand commandB = command;
798     commandB.CommandText = " INSERT INTO CITY(city_id, city_name,
                                country_id) ";
799     commandB.CommandText += "SELECT  @cityId, @cityNam,
                                COUNTRY.country_id ";
800     commandB.CommandText += "FROM COUNTRY WHERE COUNTRY.country_name =
                                @countryNam;";
801
802     commandB.Parameters.AddWithValue("@cityId", incrementCityId);
803     commandB.Parameters.AddWithValue("@countryNam", guestcountry);
804     commandB.Parameters.AddWithValue("@cityNam", guestcity);
805
806     connectionB.Open();
807     commandB.ExecuteReader();
808     connectionB.Close();
809 }
810
811
812 private void insertAddress(MySqlConnection connection, MySqlCommand
                                command, int incrementAddressId, string add, string city)
813 {
814     MySqlConnection connectionD = connection;
815     MySqlCommand commandD = command;
816     commandD.CommandText = "INSERT INTO ADDRESS(address_id,
                                address_street_number, city_id) ";
817     commandD.CommandText += "SELECT @addressId, @address, CITY.city_id ";
818     commandD.CommandText += "FROM CITY WHERE CITY.city_name =
                                @city_name;";
819
820     commandD.Parameters.AddWithValue("@addressId", incrementAddressId);
821     commandD.Parameters.AddWithValue("@address", add);
822     commandD.Parameters.AddWithValue("@city_name", city);
823
824     connectionD.Open();
825     commandD.ExecuteReader();
826     connectionD.Close();
827 }
828
829 private void insertPassenger(MySqlConnection connection, MySqlCommand
                                command, int incrementPassengerId, string guestfirstNam, string
                                guestlastNam, int getLastPhoneId, int getLastAddId, string guestemail,
                                string dateToSqlGuest)
830 {
831     MySqlConnection connectionC = connection;
832     MySqlCommand commandC = command;
833     commandC.CommandText = "INSERT INTO PASSENGER(pass_id,
                                pass_firstname, pass_lastname, pass_phone, pass_email,
                                pass_address, pass_dob) ";
834     commandC.CommandText += "values ('" + incrementPassengerId + "','" +
                                guestfirstNam + "','" + guestlastNam + "','" + getLastPhoneId +
                                "','" + guestemail + "','" + getLastAddId + "','" + dateToSqlGuest
                                + "') ";

```

```
835
836         connectionC.Open();
837         commandC.ExecuteReader();
838         connectionC.Close();
839     }
840
841     public int getLastPK(string tableName, string fieldName)
842     {
843         MySqlConnection connection;
844         MySqlCommand command = new MySqlCommand();
845         string server = "ec2-54-226-9-216.compute-1.amazonaws.com";
846         string db = "f2016_s1_user16";
847         string id = "f2016_s1_user16";
848         string pass = "f2016_s1_user16";
849         string port = "3306";
850
851         string connectionString = "SERVER=" + server + ";PORT=" + port +
            ";DATABASE=" + db + ";UID=" + id + ";PASSWORD=" + pass + ";";
852         connection = new MySqlConnection(connectionString);
853
854         connection.Open();
855
856         int lastPK = 0;
857         command.CommandText = "SELECT " + fieldName + " FROM " + tableName +
            " ORDER BY " + fieldName + " DESC LIMIT 1;";
858         command.Connection = connection;
859         MySqlDataReader sRead = command.ExecuteReader();
860         while (sRead.Read())
861         {
862             lastPK = Int16.Parse(sRead[0].ToString());
863         }
864         sRead.Close();
865
866         return lastPK;
867
868     //}
869     }
870     private void btnExit_Click(object sender, EventArgs e)
871     {
872         Form1 nextForm = new Form1();
873         this.Hide();
874         nextForm.ShowDialog();
875         this.Close();
876     }
877 }
878 }
879
```