

```

01 package GUILayer;
02
03 import ControllLayer.*;
04 import ModelLayer.*;
05
06 import java.awt.BorderLayout;
07 import java.awt.Component;
08 import java.awt.EventQueue;
09 import java.awt.Color;
10
11 import javax.swing.JFrame;
12 import javax.swing.JPanel;
13 import javax.swing.JScrollPane;
14 import javax.swing.JTable;
15 import javax.swing.ListSelectionModel;
16 import javax.swing.border.EmptyBorder;
17 import javax.swing.JTextField;
18 import javax.swing.JLabel;
19 import javax.swing.border.LineBorder;
20 import javax.swing.border.TitledBorder;
21 import javax.swing.table.DefaultTableModel;
22 import javax.swing.table.TableCellRenderer;
23
24 import java.util.ArrayList;
25 import java.util.Collection;
26 import java.util.Iterator;
27 import javax.swing.JButton;
28 import java.awt.event.MouseAdapter;
29 import java.awt.event.MouseEvent;
30
31 public class GUITransfer extends JFrame {
32
33     private static final long serialVersionUID = 1L;
34
35     private JPanel    mainPane;
36     private JPanel    backPane;
37     private JPanel    transferPane;
38     private JPanel    reportedPane;
39
40     private JTextField cageno_transferfield;
41     private JTextField cageno_reportfield;
42     private JTextField disease_field;
43     private JTextField employee_field;
44     private JTextField date_field;
45     private JTextField status_field;
46
47     JButton SaveReportButton;
48     JButton CancelReportButton;
49     JButton SaveTransferButton;
50     JButton CancelTransferButton;
51     JButton SearchTransferButton;
52     JButton DeleteTransferButton;
53
54     JLabel lblTransferdato;
55
56     int diseaseId = 0;
57
58
59     /**
60      * Launch the application.
61      */
62     public static void main(String[] args) {
63         EventQueue.invokeLater(new Runnable() {
64             public void run() {

```

```

65         try {
66             GUITransfer frame = new GUITransfer();
67             frame.setVisible(true);
68         } catch (Exception e) {
69             e.printStackTrace();
70         }
71     }
72     });
73 }
74
75 /**
76  * Create the main frame.
77  */
78 public GUITransfer() {
79     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
80     setBounds(100, 100, 1024, 600);
81     mainPane = new JPanel();
82     mainPane.setBorder(new EmptyBorder(5, 15, 5, 5));
83     mainPane.setLayout(new BorderLayout(0, 0));
84     setContentPane(mainPane);
85
86     JLabel lblDiceaseTransfereMenu = new JLabel("Rapport og
87     Transfer Vindue");
88     mainPane.add(lblDiceaseTransfereMenu, BorderLayout.NORTH);
89
90     backPane = new JPanel();
91     backPane.setBorder(new LineBorder(new Color(0, 0, 0)));
92     mainPane.add(backPane, BorderLayout.CENTER);
93     backPane.setLayout(null);
94
95     createTransferPane();
96     createReportedPane();
97
98     // Status
99     JLabel lblStatus = new JLabel("Status:");
100    lblStatus.setBounds(10, 500, 46, 14);
101    backPane.add(lblStatus);
102
103    refreshScreen();
104
105    status_field = new JTextField();
106    status_field.setText("test");
107    status_field.setEditable(false);
108    status_field.setBounds(60, 497, 916, 20);
109    backPane.add(status_field);
110    status_field.setColumns(10);
111
112 }
113
114 /**
115  * Create the Transfer frame.
116  */
117 private void createTransferPane() {
118     transferPane = new JPanel();
119     transferPane.setBorder(new TitledBorder(null, "Transfer
120     Vindue", TitledBorder.LEADING, TitledBorder.TOP, null,
121     null));
122     transferPane.setBounds(496, 10, 500, 490);
123     transferPane.setBounds(500, 10, 480, 480);
124     backPane.add(transferPane);
125     transferPane.setLayout(null);
126
127     // CageNo
128     JLabel lblNewLabel_1 = new JLabel("Indtast Bur Nummer:");

```

```

129 lblNewLabel_1.setBounds(10, 71, 137, 14);
130 lblNewLabel_1.setEnabled(true);
131 transferPane.add(lblNewLabel_1);
132
133 cageno_transferfield = new JTextField();
134 cageno_transferfield.setEnabled(false);
135 cageno_transferfield.setEditable(false);
136 cageno_transferfield.setBounds(147, 68, 86, 20);
137 transferPane.add(cageno_transferfield);
138 cageno_transferfield.setColumns(10);
139 transferPane.setVisible(true);
140
141 // Disease Id
142 JLabel lblNewLabel = new JLabel("Indtast Sygdom Id :");
143 lblNewLabel.setBounds(10, 96, 137, 14);
144 transferPane.add(lblNewLabel);
145
146 disease_field = new JTextField();
147 disease_field.setEditable(false);
148 disease_field.setEnabled(false);
149 disease_field.setBounds(147, 96, 86, 20);
150 transferPane.add(disease_field);
151 disease_field.setColumns(10);
152
153 // Employee Id
154 JLabel lblEmployeeId = new JLabel("Indtast Medarbejder Id:");
155 lblEmployeeId.setBounds(10, 133, 137, 14);
156 transferPane.add(lblEmployeeId);
157
158 employee_field = new JTextField();
159 employee_field.setBounds(147, 130, 86, 20);
160 employee_field.setEditable(false);
161 employee_field.setEnabled(false);
162 employee_field.setEditable(false);
163 transferPane.add(employee_field);
164 employee_field.setColumns(10);
165
166 lblTransferdato = new JLabel("TransferDato:");
167 lblTransferdato.setVisible(false);
168 lblTransferdato.setBounds(10, 167, 114, 14);
169 transferPane.add(lblTransferdato);
170
171 date_field = new JTextField();
172 date_field.setEditable(false);
173 date_field.setVisible(false);
174 date_field.setBounds(147, 161, 86, 20);
175 transferPane.add(date_field);
176 date_field.setColumns(10);
177
178 JButton btnNewButton_3 = new JButton("Starte Ny");
179 btnNewButton_3.addMouseListeners(new MouseAdapter() {
180     @Override
181     public void mouseClicked(MouseEvent e) {
182         cageno_transferfield.setText("");
183         disease_field.setText("");
184         employee_field.setText("");
185
186         cageno_transferfield.setEditable(true);
187         cageno_transferfield.setEnabled(true);
188         disease_field.setEditable(true);
189         disease_field.setEnabled(true);
190         employee_field.setEnabled(true);
191         employee_field.setEditable(true);
192

```

```

193
194         SaveTransferButton.setEnabled(true);
195         CancelTransferButton.setEnabled(true);
196         SearchTransferButton.setEnabled(false);
197         DeleteTransferButton.setEnabled(false);
198
199         status_field.setForeground(Color.BLACK);
200         status_field.setText("");
201
202     }
203 });
204
205 btnNewButton_3.setBounds(10, 24, 154, 23);
206 transferPane.add(btnNewButton_3);
207
208 JButton btnFindTransfer = new JButton("Find Transfer");
209 btnFindTransfer.addMouseListener(new MouseAdapter() {
210     @Override
211     public void mouseClicked(MouseEvent arg0) {
212         cageno_transferfield.setText("");
213         disease_field.setText("");
214         employee_field.setText("");
215         date_field.setText("");
216
217         cageno_transferfield.setEditable(true);
218         cageno_transferfield.setEnabled(true);
219
220         SaveTransferButton.setEnabled(false);
221         CancelTransferButton.setEnabled(false);
222         SearchTransferButton.setEnabled(true);
223         DeleteTransferButton.setEnabled(false);
224
225         status_field.setText("");
226
227     }
228 });
229
230 btnFindTransfer.setBounds(174, 24, 147, 23);
231 transferPane.add(btnFindTransfer);
232
233 SaveTransferButton = new JButton("Gem");
234 SaveTransferButton.addMouseListener(new MouseAdapter() {
235     @Override
236     public void mouseClicked(MouseEvent e) {
237         cageno_transferfield.setEditable(false);
238         cageno_transferfield.setEnabled(false);
239         disease_field.setEditable(false);
240         disease_field.setEnabled(false);
241         employee_field.setEnabled(false);
242         employee_field.setEditable(false);
243
244         SaveTransferButton.setEnabled(false);
245         CancelTransferButton.setEnabled(false);
246         SearchTransferButton.setEnabled(false);
247         DeleteTransferButton.setEnabled(false);
248
249         insertTransfer();
250         refreshScreen();
251
252         cageno_transferfield.setText("");
253         disease_field.setText("");
254         employee_field.setText("");
255         date_field.setText("");
256

```

```

257     }
258   });
259
260   SaveTransferButton.setBounds(35, 192, 89, 23);
261   SaveTransferButton.setEnabled(false);
262   transferPane.add(SaveTransferButton);
263
264   CancelTransferButton = new JButton("Annullere");
265   CancelTransferButton.addMouseListener(new MouseAdapter()
266   {
267     @Override
268     public void mouseClicked(MouseEvent e) {
269       cageno_transferfield.setEditable(false);
270       cageno_transferfield.setEnabled(false);
271       disease_field.setEditable(false);
272       disease_field.setEnabled(false);
273       employee_field.setEnabled(false);
274       employee_field.setEditable(false);
275       date_field.setVisible(false);
276       lblTransferdato.setVisible(false);
277
278       SaveTransferButton.setEnabled(false);
279       CancelTransferButton.setEnabled(false);
280       SearchTransferButton.setEnabled(false);
281       DeleteTransferButton.setEnabled(false);
282
283       cageno_transferfield.setText("");
284       disease_field.setText("");
285       employee_field.setText("");
286       date_field.setText("");
287     }
288   });
289
290   CancelTransferButton.setBounds(134, 192, 89, 23);
291   CancelTransferButton.setEnabled(false);
292   transferPane.add(CancelTransferButton);
293
294   SearchTransferButton = new JButton("S\u00F8g");
295   SearchTransferButton.setEnabled(false);
296   SearchTransferButton.addMouseListener(new MouseAdapter()
297   {
298     @Override
299     public void mouseClicked(MouseEvent e) {
300       if((cageno_transferfield.getText().length() > 2) &&
301         findTransfer())
302       {
303         status_field.setForeground(Color.BLACK);
304         status_field.setText("KLAR TIL AT SLETTE VIST
305           TRANSFER POST.");
306         DeleteTransferButton.setEnabled(true);
307       }
308       else
309       {
310         status_field.setForeground(Color.RED);
311         status_field.setText("FEJL BUR NUMMER MANGLER AT
312           BLIVE ANGIVET ELLER FINDES IKKE I TRANSFER.");
313       }
314       SaveTransferButton.setEnabled(false);
315       CancelTransferButton.setEnabled(true);
316       SearchTransferButton.setEnabled(false);
317
318       disease_field.setEnabled(true);
319       employee_field.setEnabled(true);
320       date_field.setVisible(true);

```

```

321         lblTransferdato.setVisible(true);
322     }
323 }
324 });
325 SearchTransferButton.setBounds(266, 192, 89, 23);
326 transferPane.add(SearchTransferButton);
327
328 DeleteTransferButton = new JButton("Slet");
329 DeleteTransferButton.addMouseListener(new MouseAdapter()
330 {
331     @Override
332     public void mouseClicked(MouseEvent e) {
333         if(DeleteTransferButton.isEnabled())
334         {
335             cageno_transferfield.setEditable(false);
336             cageno_transferfield.setEnabled(false);
337             disease_field.setEnabled(false);
338             employee_field.setEnabled(false);
339             date_field.setVisible(false);
340             lblTransferdato.setVisible(false);
341
342             SaveTransferButton.setEnabled(false);
343             CancelTransferButton.setEnabled(false);
344             SearchTransferButton.setEnabled(false);
345             DeleteTransferButton.setEnabled(false);
346
347             deleteTransfer();
348             //createTransferList();
349
350             cageno_transferfield.setText("");
351             disease_field.setText("");
352             employee_field.setText("");
353             date_field.setText("");
354         }
355     }
356
357     createTransferList();
358 }
359 });
360 DeleteTransferButton.setEnabled(false);
361 DeleteTransferButton.setBounds(365, 192, 89, 23);
362 transferPane.add(DeleteTransferButton);
363
364 }
365
366
367 // method to insert a Transfer record into the database
368 private void insertTransfer()
369 {
370     CTRTransfer ctrtrans = new CTRTransfer();
371     int cagenumber = Integer.parseInt(cageno_transferfield.
372     getText());
373     int disease = Integer.parseInt(disease_field.getText());
374     int employee = Integer.parseInt(employee_field.getText());
375
376     if((checkCage(cagenumber) != null) &&
377        (checkEmployee(employee) != null) &&
378        ((checkBite(disease) != null) || (checkPlasma(
379        disease) != null))
380    )
381    {
382        if(ctrtrans.insertTransfer(cagenumber, disease,
383        employee) == 1)
384        {

```

```

385         status_field.setForeground(Color.BLACK);
386         status_field.setText("NY TRANSFER OPRETTET.");
387     }
388     else
389     {
390         status_field.setForeground(Color.RED);
391         status_field.setText("EN FEJL OPSTOD UNDER OPRETTELSE
392         AF TRANSFER. SE KONSOL VINDUE.");
393     }
394 }
395 }
396 else
397 {
398     status_field.setForeground(Color.RED);
399     status_field.setText("FEJL I EN ELLER FLERE NØGLER.
400     TRANSFER IKKE OPRETTET. SE KONSOL VINDUE.");
401 }
402 }
403
404 private void deleteTransfer()
405 {
406     CTRTransfer ctrtrans = new CTRTransfer();
407     int cagenumber = Integer.parseInt(cageno_transferfield.
408     getText());
409     if(ctrtrans.deleteTransfer(cagenumber) == 1)
410     {
411         status_field.setForeground(Color.BLACK);
412         status_field.setText("TRANSFER POST SLETTET.");
413     }
414     else
415     {
416         status_field.setForeground(Color.RED);
417         status_field.setText("EN FEJL OPSTOD UNDER SLETNINGEN
418         AF TRANSFER. SE KONSOL VINDUE.");
419     }
420 }
421 }
422 }
423
424 // method to create a scroll list of all Transfers.
425 private void createTransferList()
426 {
427     CTRTransfer ctrTransfer = new CTRTransfer();
428     ArrayList<Transfer> TransferList = ctrTransfer.
429     getAllTransfer();
430
431     Collection<Transfer> TransferColList = TransferList;
432     Iterator<Transfer> itr = TransferColList.iterator();
433     int itemSize = TransferColList.size();
434     String tabledata[][] = new String[itemSize][5];
435     int i = 0;
436     while( itr.hasNext() ) {
437
438         Transfer TransferObj = itr.next();
439         Integer TransferIdInt = TransferObj.getTransferId();
440         Integer CageNoInt = TransferObj.getCage().getCageNo();
441         Integer DiseaseIdInt;
442         if(TransferObj.getBitedisease() != null)
443             DiseaseIdInt = TransferObj.getBitedisease().
444             getDiseaseId();
445         else if (TransferObj.getPlasmadisease() != null)
446             DiseaseIdInt = TransferObj.getPlasmadisease().
447             getDiseaseId();
448         else

```

```

449         DiseaseIdInt = 0;
450         Integer EmployeeIdInt = TransferObj.getEmployee().
451         getEmployeeID();
452
453         String line[] = {
454             TransferIdInt.toString(),
455             CageNoInt.toString(),
456             DiseaseIdInt.toString(),
457             EmployeeIdInt.toString(),
458             TransferObj.getTransferDate()
459         };
460
461         tabledata[i] = line;
462         i++;
463     }
464
465     String headers[] = { "Transfer ID", "Cage No", "Disease
466     ID", "Employee ID", "Transfer Date" };
467     // Create a new table instance
468     DefaultTableModel model = new DefaultTableModel(
469     tabledata, headers);
470     // And line color shift in table
471     JTable table = new JTable(model) {
472         private static final long serialVersionUID = 1L;
473
474         public Component prepareRenderer(TableCellRenderer
475         renderer, int Index_row, int Index_col) {
476             Component comp = super.prepareRenderer(renderer,
477             Index_row, Index_col);
478             // if not selected then change background color
479             for every second row
480             if (!isCellSelected(Index_row, Index_col)) {
481                 if (Index_row % 2 == 0)
482                     comp.setBackground(Color.WHITE);
483                 else
484                     comp.setBackground(Color.LIGHT_GRAY);
485             }
486             return comp;
487         }
488     }; // end of creating JTable table
489
490     table.setSelectionMode(ListSelectionModel.
491     SINGLE_SELECTION);
492     table.setEnabled(false);
493
494     JScrollPane scrollPane = new JScrollPane(table);
495     scrollPane.setViewportBorder(new TitledBorder(null,
496     "Transfer list", TitledBorder.LEADING, TitledBorder.TOP,
497     null, null));
498     scrollPane.setBounds(10, 226, 460, 244);
499     transferPane.add(scrollPane, BorderLayout.CENTER);
500
501 }
502
503
504 /**
505  * Create the DiseaseReport frame.
506  */
507 private void createReportedPane() {
508     reportedPane = new JPanel();
509     reportedPane.setBorder(new TitledBorder(null, "Rapport
510     Vindue", TitledBorder.LEADING, TitledBorder.TOP, null,
511     null));
512     reportedPane.setBounds(10, 10, 480, 480);

```



```

513     backPane.add(reportedPane);
514     reportedPane.setLayout(null);
515
516     // Cage No
517     JLabel lblNewLabel = new JLabel("Indtast Bur Nummer:");
518     lblNewLabel.setBounds(10, 113, 131, 14);
519     lblNewLabel.setEnabled(true);
520     reportedPane.add(lblNewLabel);
521
522     cageno_reportfield = new JTextField();
523     cageno_reportfield.setBounds(144, 110, 92, 20);
524     cageno_reportfield.setEnabled(false);
525     cageno_reportfield.setEditable(false);
526     reportedPane.add(cageno_reportfield);
527     cageno_reportfield.setColumns(10);
528
529     JButton btnNewButton = new JButton("Registrere Ny
530     Plasmacytosis ");
531     btnNewButton.addMouseListener(new MouseAdapter() {
532         @Override
533         public void mouseClicked(MouseEvent arg0) {
534             diseaseId = 1500;
535             cageno_reportfield.setText("");
536             cageno_reportfield.setEnabled(true);
537             cageno_reportfield.setEditable(true);
538
539             SaveReportButton.setEnabled(true);
540             CancelReportButton.setEnabled(true);
541
542             status_field.setForeground(Color.BLACK);
543             status_field.setText("KLAR TIL AT OPRETTE EN DISEASE
544             REPORT PÅ ET BUR MED SYGDOM PLASMACYTOSIS.");
545         }
546     });
547
548     btnNewButton.setBounds(10, 23, 210, 23);
549     reportedPane.add(btnNewButton);
550
551     JButton btnNewButton_1 = new JButton("Registrere Ny
552     Bides\u00E5r");
553     btnNewButton_1.addMouseListener(new MouseAdapter() {
554         @Override
555         public void mouseClicked(MouseEvent e) {
556             diseaseId = 1600;
557             cageno_reportfield.setText("");
558             cageno_reportfield.setEnabled(true);
559             cageno_reportfield.setEditable(true);
560
561             SaveReportButton.setEnabled(true);
562             CancelReportButton.setEnabled(true);
563
564             status_field.setForeground(Color.BLACK);
565             status_field.setText("KLAR TIL AT OPRETTE EN
566             DISEASEREPORT PÅ ET BUR MED SYGDOM BIDESÅR.");
567         }
568     });
569
570     btnNewButton_1.setBounds(10, 57, 173, 23);
571     reportedPane.add(btnNewButton_1);
572
573     SaveReportButton = new JButton("Gem");
574     SaveReportButton.setEnabled(false);
575     SaveReportButton.addMouseListener(new MouseAdapter() {
576         @Override

```

```

577     public void mouseClicked(MouseEvent e) {
578         cageno_reportfield.setEnabled(false);
579         cageno_reportfield.setEditable(false);
580         SaveReportButton.setEnabled(false);
581         CancelReportButton.setEnabled(false);
582
583         createDiseaseReport();
584         refreshScreen();
585
586         cageno_reportfield.setText("");
587     }
588 });
589
590 SaveReportButton.setBounds(95, 193, 83, 23);
591 reportedPane.add(SaveReportButton);
592
593 CancelReportButton = new JButton("Annullere");
594 CancelReportButton.addMouseListener(new MouseAdapter() {
595     @Override
596     public void mouseClicked(MouseEvent e) {
597         cageno_reportfield.setEnabled(false);
598         cageno_reportfield.setEditable(false);
599
600         SaveReportButton.setEnabled(false);
601         CancelReportButton.setEnabled(false);
602
603         cageno_reportfield.setText("");
604         diseaseId = 0;
605     }
606 });
607
608 CancelReportButton.setEnabled(false);
609 CancelReportButton.setBounds(188, 193, 89, 23);
610 reportedPane.add(CancelReportButton);
611
612 }
613
614
615 // method to insert a DiseaseReport record into the
616 database
617 private void createDiseaseReport()
618 {
619     CTRDiseaseReport ctrDiseaseReport = new CTRDiseaseReport();
620     int cagenumber = Integer.parseInt(cageno_reportfield.
621     getText());
622
623     if((checkCage(cagenumber) != null) &&
624         ((checkBite(diseaseId) != null) || (checkPlasma(
625         diseaseId) != null))
626     )
627     {
628         if(checkCageNoTransfer(cagenumber) == null)
629         {
630             if(ctrDiseaseReport.insertDiseaseReport(cagenumber,
631             diseaseId) == 1)
632             {
633                 status_field.setForeground(Color.BLACK);
634                 status_field.setText("NY DISEASE REPORT
635 OPRETTET.");
636             }
637             else
638             {
639                 status_field.setForeground(Color.RED);
640                 status_field.setText("FEJL OPSTOD UNDER OPRETTELSE

```

```

AF
640             DISEASE REPORT. SE KONSOL VINDUE.");
641         }
642     }
643     else
644     {
645         status_field.setForeground(Color.RED);
646         status_field.setText("FEJL BUR NUMMER EKSISTER I FORVEJN
647             I TRANSFER. DISEASE REPORT IKKE OPRETTET.");
648     }
649 }
650 else
651 {
652     status_field.setForeground(Color.RED);
653     status_field.setText("FEJL I EN ELLER FLERE NØGLER.
654         DISEASE REPORT IKKE OPRETTET. SE KONSOL VINDUE.");
655 }
656 }
657
658
659 // method to create a scroll list of all DiseaseReports.
660 private void createReportList()
661 {
662     //HashMap<Integer, Transfer> TransferList;
663     //TransferList = new HashMap<Integer, Transfer>();
664     CTRDiseaseReport ctrDiseaseReport = new CTRDiseaseReport();
665     ArrayList<DiseaseReport> DiseaseReportList;
666
667     DiseaseReportList = ctrDiseaseReport.getAllDiseaseReport();
668     Collection<DiseaseReport> DiseaseReportColList =
669         DiseaseReportList;
670     Iterator<DiseaseReport> itr = DiseaseReportColList.
671         iterator();
672     int itemSize = DiseaseReportColList.size();
673     String tabledata[][] = new String[itemSize][3];
674     int i = 0;
675     while( itr.hasNext() ) {
676
677         DiseaseReport DiseaseReportObj = itr.next();
678         Integer CageNoInt = DiseaseReportObj.getCageNo();
679         Integer DiseaseIdInt = DiseaseReportObj.getDiseaseId();
680
681         String line[] = {
682             CageNoInt.toString(),
683             DiseaseIdInt.toString(),
684             DiseaseReportObj.getReportDate()
685         };
686
687         tabledata[i] = line;
688         i++;
689     }
690
691     String headers[] = { "Cage No", "Disease ID", "Report
692         Date" };
693     // Create a new table instance
694     DefaultTableModel model = new DefaultTableModel(
695         tabledata, headers);
696     // And line color shift in table
697     JTable table = new JTable(model) {
698         private static final long serialVersionUID = 1L;
699
700         public Component prepareRenderer(TableCellRenderer
701             renderer, int Index_row, int Index_col) {
702             Component comp = super.prepareRenderer(renderer,

```

```

703         Index_row, Index_col);
704         // if not selected then change background color
705         for every second row
706         if (!isCellSelected(Index_row, Index_col)) {
707             if (Index_row % 2 == 0)
708                 comp.setBackground(Color.WHITE);
709             else
710                 comp.setBackground(Color.LIGHT_GRAY);
711         }
712         return comp;
713     }
714 }; // end of creating JTable table
715
716 table.setSelectionMode(ListSelectionModel.
717 SINGLE_SELECTION);
718 table.setEnabled(false);
719
720 JScrollPane scrollPane = new JScrollPane(table);
721 scrollPane.setViewportBorder(new TitledBorder(null,
722 "DiseaseReport list", TitledBorder.LEADING,
723 TitledBorder.TOP, null, null));
724 scrollPane.setBounds(10, 227, 460, 243);
725 reportedPane.add(scrollPane, BorderLayout.CENTER);
726
727 }
728
729 // method to refresh the two list tables after an operation.
730 private void refreshScreen()
731 {
732     createReportList();
733     createTransferList();
734 }
735
736 //method to find cage
737 private Cage checkCage(int key)
738 {
739     CTCage ctrEmp = new CTCage();
740     return ctrEmp.findCage(key);
741 }
742
743 //method to find employee
744 private Employee checkEmployee(int key)
745 {
746     CTREmployee ctrEmp = new CTREmployee();
747     return ctrEmp.findEmployeeById(key);
748 }
749
750 //method to find bitedisease
751 private BiteDisease checkBite(int key)
752 {
753     CTRBiteDisease ctrBite = new CTRBiteDisease();
754     return ctrBite.findDiseaseById(key);
755 }
756
757 //method to find plasmadisease
758 private PlasmaDisease checkPlasma(int key)
759 {
760     CTRPlasmaDisease ctrplasma = new CTRPlasmaDisease();
761     return ctrplasma.findDiseaseById(key);
762 }
763
764 //method to find Cage number in Transfer
765 private Transfer checkCageNoTransfer(int key)
766 {

```

```

767         CTRTransfer ctrtransfer = new CTRTransfer();
768         return ctrtransfer.searchTransfersByCageId(key);
769     }
770
771     //method a Transfer with given Cage number and put it into
772     display fields.
773     private boolean findTransfer()
774     {
775         if(cageno_transferfield.getText().length() > 2)
776         {
777             int cagenumber = Integer.parseInt(cageno_transferfield.
778             getText());
779             Transfer transfer = checkCageNoTransfer(cagenumber);
780             if(transfer != null)
781             {
782                 employee_field.setText(String.valueOf(transfer.
783                 getEmployee().getEmployeeID()));
784                 if(transfer.getBitedisease() != null)
785                     disease_field.setText(String.valueOf(transfer.
786                     getBitedisease().getDiseaseId()));
787                 else if(transfer.getPlasmadisease() != null)
788                     disease_field.setText(String.valueOf(transfer.
789                     getPlasmadisease().getDiseaseId()));
790                 else
791                     disease_field.setText("NOT GIVEN");
792                 date_field.setText(transfer.getTransferDate());
793                 return true;
794             }
795         }
796         return false;
797     }
798
799 }

```