

```

//importend libraries
import java.util.ArrayList;
import java.util.Scanner;
import classes.*;

//Booking App class
1 public class BookingApp {
2
3     //the mainMenu method call mainMenu
4     public static String mainMenu(){
5
6         //the format of the the main Menu
7         System.out.println("University of Knowledge - COVID test");
8         System.out.println();
9         System.out.println("Manage Bookings");
10        System.out.println();
11        System.out.println("Please, enter the number to select your option: ");
12        System.out.println();
13        System.out.println("To manage Bookable Rooms: ");
14        System.out.println(" 1. List ");
15        System.out.println(" 2. Add ");
16        System.out.println(" 3. Remove ");
17        System.out.println("To manage Assistants on Shift: ");
18        System.out.println(" 4. List ");
19        System.out.println(" 5. Add ");
20        System.out.println(" 6. Remove ");
21        System.out.println("To manage Bookings: ");
22        System.out.println(" 7. List");
23        System.out.println(" 8. Add ");
24        System.out.println(" 9. Remove ");
25        System.out.println(" 10. Conclude ");
26        System.out.println("After selecting one the options above, you will be presented other screens.");
27        System.out.println("If you press 0, you will be able to return to this main menu.");
28        System.out.println("Press -1 (or ctrl+c) to quit this application.");
29        System.out.println("");
30
31        //scanner for user input
32        Scanner sc = new Scanner(System.in); //System.in is a standard input stream
33        String user_input = sc.nextLine(); //reads string
34
35        //this method will return the input from the user
36        return user_input;
37    }
38
39
40 /**
41  * @param args
42  */
43 public static void main(String[] args){
44     //flag for calling the main menu
45     boolean flag = true;
46
47     //initializing system
48     //rooms
49     Room room1 = new Room("A1",3);
50     Room room2 = new Room("B23", 5);
51     Room room3 = new Room("C13", 2);
52     //array of rooms
53     Room[] rooms = {room1,room2,room3};
54     //populate Univercity Resources
55     UniversityResources.addRooms(rooms);
56
57     //assistants
58     Assistant as1 = new Assistant("jo97@uok.ac.uk", "John");
59     Assistant as2 = new Assistant("cate123@uok.ac.uk", "Kate");
60
61     Assistant as3 = new Assistant("geog2000@uok.ac.uk", "George");
62     //array of assitants
63     Assistant[] assistants = {as1,as2,as3};
64     //populate Univercity Resources
65     UniversityResources.addAssistans(assistans);
66
67     //bookable rooms

```

```

68 BookableRoom br1 = new BookableRoom(1, "07:00", "25/02/2021", room1); // Available
69 BookableRoom br2 = new BookableRoom(3, "08:00", "26/02/2021", room1); // Full
70 BookableRoom br3 = new BookableRoom(1, "09:00", "27/02/2021", room1); // Available
71 BookableRoom br4 = new BookableRoom(4, "07:00", "26/02/2021", room2); // Available
72 BookableRoom br5 = new BookableRoom(1, "08:00", "28/02/2021", room2); // Available
73 BookableRoom br6 = new BookableRoom(0, "09:00", "26/02/2021", room2); // Empty
74 BookableRoom br7 = new BookableRoom(2, "07:00", "23/02/2021", room3); // Full
75 BookableRoom br8 = new BookableRoom(1, "08:00", "23/02/2021", room3); // Available
76 BookableRoom br9 = new BookableRoom(0, "09:00", "21/02/2021", room3); // Empty
77 //array of bookable rooms
78 BookableRoom[] bookablerooms = {br1,br2,br3,br4,br5,br6,br7,br8,br9};
79 //passing them to the bookable room array list in Booking System
80 BookingSystem.populateBookableRooms(bookablerooms);
81
82 //assistant on shift
83 AssistantOnShift asos1 = new AssistantOnShift("25/02/2021", "FREE", as1); // Free
84 AssistantOnShift asos2 = new AssistantOnShift("21/02/2021", "BUSY", as1); // Busy
85 AssistantOnShift asos3 = new AssistantOnShift("23/02/2021", "FREE", as2); // Free
86 AssistantOnShift asos4 = new AssistantOnShift("28/02/2021", "FREE", as2); // Free
87 AssistantOnShift asos5 = new AssistantOnShift("23/02/2021", "BUSY", as3); // Busy
88 AssistantOnShift asos6 = new AssistantOnShift("26/02/2021", "FREE", as3); // Free
89 //array of assistants on shift
90 AssistantOnShift[] assistantonshifts = {asos1,asos2,asos3,asos4,asos5,asos6};
91 //passing them to the bookable room array list in Booking System
92 BookingSystem.populateAssistantOnShifts(assistantonshifts);
93
94
95 //main while loop for calling the main menu
96 while (flag == true) {
97     //calling main menu
98     String user_input = BookingApp.mainMenu();
99
100     //if -1 is pressed then systme exists
101     if (user_input.equals("-1")) {
102         System.out.println("Quit");
103         System.exit(0);
104     }
105
106     //if 1 is pressed we go to the next page
107     if (user_input.equals("1")) {
108         //format of the 1st page
109         System.out.println("University of Knowledge - COVID test");
110         System.out.println();
111
112         //print all bookable rooms
113         for (BookableRoom br:BookingSystem.getBookableRooms()){
114             br.printThe();
115         }
116
117         //formatting the rest of the page
118         System.out.println("0. Back to main menu.");
119         System.out.println("-1. Quit application.");
120         System.out.println();
121
122         //flag to continue calling until 0 or -1 pressed
123         boolean f = true;
124         //loop for calls
125         while (f == true){
126             //getting users input
127             Scanner sc = new Scanner(System.in);
128             String us_in = sc.nextLine();
129
130             //if 0 then exit from page
131             if (us_in.equals("0")){
132                 f = false;
133                 //if -1 then exit from the system
134             } else if (us_in.equals("-1")){
135                 System.exit(0);
136                 //otherwise an error
137             } else {
138                 System.out.println("Error! Please enter a vaild input");
139                 flag = true;
140             }
141

```

```

142     }
143 }
144
145 //second page
146 if (user_input.equals("2")){
147     //format of this page
148     System.out.println("University of Knowledge - COVID test");
149     System.out.println();
150     System.out.println("Adding bookable room");
151     System.out.println();
152     System.out.println(" Please, enter one of the following: ");
153     System.out.println();
154
155     //printnig out all rooms of the system
156     int k = 1;
157     for (Room r : rooms) {
158         System.out.print(k + ". ");
159         r.printThe();
160         k = k + 1;
161     }
162
163     //rest of the format of the page
164     System.out.println("The sequential ID listed to a room, a date (dd/mm/yyyy), and a time (HH:MM),");
165     System.out.println("separated by a white space.");
166     System.out.println("0. Back to main menu.");
167     System.out.println("-1. Quit application.");
168     System.out.println();
169
170     //flag for continue calling
171     boolean new_flag = true;
172
173     //the loop with the flag
174     while (new_flag == true) {
175
176         //getting user input
177         Scanner sc = new Scanner(System.in);
178         String[] us_in = sc.nextLine().split(" ");
179
180         //exit to main menu
181         if (us_in[0].equals("0")){
182             new_flag = false;
183
184         //exit the system
185         } else if (us_in[0].equals("-1")){
186             System.exit(0);
187         }
188
189         //try to add a bookable room
190         try {
191             //if statment to control user input
192             if (new_flag == true){
193                 //converting the id to int
194
195                 int i = Integer.parseInt(us_in[0]);
196
197                 //validating users input
198                 BookingSystem.validateTime(us_in[2]);
199                 BookingSystem.validateDate(us_in[1]);
200
201                 //adding the room
202                 //if invalid data is entered, stystem prints out the catch
203                 BookingSystem.addBookableRoom(0, us_in[2], us_in[1], rooms[i-1]);
204
205                 //formating of the suuccess page
206                 System.out.println("Bookable Room added successfully");
207                 //printing out the new bookable room
208                 System.out.println(BookingSystem.getBookableRooms().get(BookingSystem.getBookableRooms().size()-1).printThe());
209                 System.out.println("Please, enter one of the above!");
210                 System.out.println();
211                 System.out.println("The sequential ID listed to a room, a date (dd/mm/yyyy), and a time (HH:MM),");
212                 System.out.println("separated by a white space.");
213                 System.out.println("0. Back to main menu.");
214                 System.out.println("-1. Quit application.");
215                 System.out.println();

```

```

215         System.out.println();
216     }else{
217         System.out.println();
218     }
219
220     //catch if an error occurs
221 } catch(Exception e) {
222     //format of a fail page
223     System.out.println("Error!");
224     System.out.println("This input is invalid, please follow the correct syntax");
225     System.out.println("Please, enter one of the above!");
226     System.out.println();
227     System.out.println("The sequential ID listed to a room, a date (dd/mm/yyyy), and a time (HH:MM),");
228     System.out.println("separated by a white space.");
229     System.out.println("0. Back to main menu.");
230     System.out.println("-1. Quit application.");
231     System.out.println();
232
233 }
234 }
235 }
236
237 //third page
238 if (user_input.equals("3")){
239
240     //format of the page
241     System.out.println("University of Knowledge - COVID test");
242     System.out.println();
243     System.out.println("Removing bookable room");
244     System.out.println();
245     System.out.println("Please, enter one of the following:");
246     System.out.println();
247
248     //printing out all bookable rooms that can be removed
249     for (BookableRoom br : BookingSystem.getBookableRooms()){
250         if (br.findOutStatusRoom() == "EMPTY"){
251             int indx = BookingSystem.getBookableRooms().indexOf(br);
252             System.out.print(indx + 1 + ". ");
253             System.out.println(br.printThe());
254         }
255     }
256
257     //rest of the formatting
258     System.out.println("The sequential ID listed to a room, a date (dd/mm/yyyy), and a time (HH:MM),");
259     System.out.println("0. Back to main menu.");
260     System.out.println("-1. Quit application.");
261     System.out.println();
262
263     //flag for continuous calls
264     boolean new_new_flag = true;
265
266     //loop with the flag
267     while (new_new_flag == true){
268
269         //getting user input
270         Scanner sc = new Scanner(System.in);
271         String us_in = sc.nextLine();
272
273         //exit to main menu
274         if (us_in.equals("0")){
275             new_new_flag = false;
276
277             //exit the system
278         } else if (us_in.equals("-1")){
279             System.exit(0);
280         }
281
282         //try to remove the bookable room
283         try{
284             //if statement to control users input
285             if (new_new_flag == true) {
286                 //converting user input to integer
287                 int i = Integer.parseInt(us_in);

```

```

289         //format of the success page
290         System.out.println("Bookable Room removed successfully:");
291         //printing the element which will be removed
292         System.out.println(BookingSystem.getBookableRooms().get(i-1).printThe());
293
294         //removing the room which was picked
295         BookingSystem.removeBookableRoom(BookingSystem.getBookableRooms().get(i-1));
296
297         //rest of the format
298         System.out.println("Please, enter one of the above!");
299         System.out.println();
300         System.out.println("The sequential ID to select the bookable room to be removed.");
301         System.out.println("0. Back to main menu.");
302         System.out.println("-1. Quit application.");
303         System.out.println();
304     } else{
305         System.out.println();
306     }
307
308     // if error occurs
309 } catch (Exception e){
310     //format of the fail page
311     System.out.println("Error!");
312     System.out.println("This input is invaild, please follow the correct syntax");
313     System.out.println("Please, enter one of the above!");
314     System.out.println();
315     System.out.println("The sequential ID to select the bookable room to be removed.");
316     System.out.println("0. Back to main menu.");
317     System.out.println("-1. Quit application.");
318     System.out.println();
319 }
320
321 }
322
323 }
324
325 //forth page
326 if (user_input.equals("4")){
327
328     //format of the page
329
330     System.out.println("University of Knowledge - COVID test");
331     System.out.println();
332
333     //printing the assistants on shift
334     for (AssistantOnShift asos: BookingSystem.getAssistantOnShifts()){
335         asos.printThe();
336     }
337
338     //rest of the format
339     System.out.println("0. Back to main menu.");
340     System.out.println("-1. Quit application.");
341     System.out.println();
342
343     //flag for continious calls
344     boolean fla = true;
345     while (fla == true){
346
347         //getting user input
348         Scanner sc = new Scanner(System.in);
349         String us_in = sc.nextLine();
350
351         //exit to main menu
352         if (us_in.equals("0")){
353             fla = false;
354             //exit the system
355         } else if (us_in.equals("-1")){
356             System.exit(0);
357             //otherwise error
358         } else {
359             System.out.println("Error! Please enter a vaild input");
360             fla = true;
361         }
362     }

```

```

363 }
364
365 //fifth page
366 if (user_input.equals("5")){
367
368     //format of the page
369     System.out.println("University of Knowledge - COVID test");
370     System.out.println();
371     System.out.println("Adding assistant on shift");
372     System.out.println();
373     System.out.println("Please, enter one of the following: ");
374     System.out.println();
375
376     //printing all assistants of the system
377     int k = 1;
378     for (Assistant a : assistants) {
379         System.out.print(k + ". ");
380         System.out.println(a.printThe());
381         k = k + 1;
382     }
383
384
385     //rest format of the page
386     System.out.println("The sequential ID of an assistant and date (dd/mm/yyyy), separated by a white space.");
387     System.out.println("0. Back to main menu.");
388     System.out.println("-1. Quit application.");
389     System.out.println();
390
391     //flag for continuous calls
392     boolean new_flg = true;
393
394     //loop with the flag
395
396     while (new_flg == true) {
397
398         //getting user input
399         Scanner sc = new Scanner(System.in);
400         String[] us_in = sc.nextLine().split(" ");
401
402         //exit to main menu
403         if (us_in[0].equals("0")){
404             new_flg = false;
405
406         //exit the system
407         } else if (us_in[0].equals("-1")){
408             System.exit(0);
409         }
410
411         //try to add assistant on shift
412         try{
413             //if statement to control user input
414             if (new_flg == true){
415                 //converting the user input
416                 int i = Integer.parseInt(us_in[0]);
417
418                 //validating the user input (date)
419                 BookingSystem.validateDate(us_in[1]);
420
421                 //adding the assistant on shift
422                 BookingSystem.addAssistanOnShift(us_in[1], "FREE", assistants[i-1]);
423
424                 //format of the page
425                 System.out.println("Assistant on Shift added successfully.");
426
427                 //print the assistant on shift
428                 System.out.println(BookingSystem.getAssistantOnShifts().get(BookingSystem.getAssistantOnShifts().size()-1).printThe());
429                 System.out.println("Please, enter one of the above!");
430                 System.out.println();
431                 System.out.println("The sequential ID of an assistant and date (dd/mm/yyyy), separated by a white space.");
432                 System.out.println("0. Back to main menu.");
433                 System.out.println("-1. Quit application.");
434                 System.out.println();
435             } else {
436

```

```

437         System.out.println();
438     }
439     //if error occures
440     } catch (Exception e){
441         //format of the fail page
442         System.out.println("Error!");
443         System.out.println("This input is invaild, please follow the correct syntax");
444         System.out.println("Please, enter one of the above!");
445         System.out.println();
446         System.out.println("The sequential ID of an assistant and date (dd/mm/yyyy), separated by a white space.");
447         System.out.println("0. Back to main menu.");
448         System.out.println("-1. Quit application.");
449         System.out.println();
450     }
451 }
452 }
453 }
454
455 //sixth page
456 if (user_input.equals("6")){
457
458     //format of the page
459     System.out.println("University of Knowledge - COVID test");
460     System.out.println();
461     System.out.println("Removing assistant on shift");
462     System.out.println();
463
464     System.out.println("Please, enter one of the following:");
465     System.out.println();
466
467     //printing assistants on shift that can be removed
468     for (AssistantOnShift a : BookingSystem.getAssistantOnShifts()){
469         if (a.findOutStatusAssistan() == "FREE"){
470             int indx = BookingSystem.getAssistantOnShifts().indexOf(a);
471             System.out.print(indx + 1 + ". ");
472             a.printThe();
473         }
474     }
475
476     //rest format of the page
477     System.out.println("The sequential ID to select the bookable room to be removed.");
478     System.out.println("0. Back to main menu.");
479     System.out.println("-1. Quit application.");
480     System.out.println();
481
482     //flag for continious calls
483     boolean new_new_flg = true;
484
485     //loop with the flag
486     while (new_new_flg == true){
487
488         //getting user input
489         Scanner sc = new Scanner(System.in);
490         String us_in = sc.nextLine();
491
492         //exit to main menu
493         if (us_in.equals("0")){
494             new_new_flg = false;
495
496         }
497
498         //exit the sytem
499         } else if (us_in.equals("-1")){
500             System.exit(0);
501         }
502
503         //try to remove assistant
504         try{
505             //if statement to control user input
506             if (new_new_flg == true){
507                 //converting user input to int
508                 int i = Integer.parseInt(us_in);
509                 //printing out the assitatnt on shif that will be removed
510                 System.out.println("Assistant on Shift removed successfully.");
511                 System.out.println(BookingSystem.getAssistantOnShifts().get(i-1).printThe());
512
513                 //removig the assitant on shift

```



```

510 BookingSystem.removeAssistanOnShift(BookingSystem.getAssistantOnShifts().get(i-1));
511
512 System.out.println("Please, enter one of the above!");
513 System.out.println();
514 System.out.println("The sequential ID to select the assistant on shift to be removed.");
515 System.out.println("0. Back to main menu.");
516 System.out.println("-1. Quit application.");
517 System.out.println();
518 }else{
519     System.out.println();
520 }
521
522 //if error occures
523 } catch (Exception e){
524     //format of the fail page
525     System.out.println("Error!");
526     System.out.println("This input is invaild, please follow the correct syntax");
527     System.out.println("Please, enter one of the above!");
528     System.out.println();
529
530     System.out.println("The sequential ID to select the assistant on shift to be removed.");
531     System.out.println("0. Back to main menu.");
532     System.out.println("-1. Quit application.");
533     System.out.println();
534 }
535
536 }
537 }
538
539 //seventh page
540 if (user_input.equals("7")){
541     //format of the page
542     System.out.println("University of Knowledge - COVID test");
543     System.out.println();
544     System.out.println("Select which booking to list:");
545     System.out.println("1. ALL");
546     System.out.println("2. Only bookings status:SCHEDULED");
547     System.out.println("3. Only bookings status:COMPLETED");
548     System.out.println("0. Back to main menu.");
549     System.out.println("-1. Quit application.");
550     System.out.println();
551
552     //flag for continious calls
553     boolean nw_flg = true;
554
555     //loop with the flag
556     while (nw_flg == true){
557
558         //getting user input
559         Scanner sc = new Scanner(System.in);
560         String us_in = sc.nextLine();
561
562         //exit to main menu
563         if (us_in.equals("0")){
564             nw_flg = false;
565
566             //exit the system
567         } else if (us_in.equals("-1")){
568             System.exit(0);
569
570             //print all scheduled bookings
571         } else if (us_in.equals("2")){
572             BookingSystem.showSCHEDULEDBookings();
573             nw_flg = true;
574             System.out.println("0. Back to main menu.");
575             System.out.println("-1. Quit application.");
576             System.out.println();
577
578             //print all completed bookings
579         } else if (us_in.equals("3")){
580             BookingSystem.showCOMPLETEDBookings();
581             nw_flg = true;
582             System.out.println("0. Back to main menu.");
583             Svsstem.out.println("-1. Quit application.");
584

```



```

584         System.out.println();
585
586         //print all bookings
587     } else {
588         BookingSystem.showALLBookings();
589         nw_flg = true;
590         System.out.println("0. Back to main menu.");
591         System.out.println("-1. Quit application.");
592         System.out.println();
593     }
594 }
595 }
596
597
598 //eighth page
599 if (user_input.equals("8")){
600     //format of the page
601     System.out.println("University of Knowledge - COVID test");
602     System.out.println();
603     System.out.println("Adding booking (appointment for a COVID test) to the system");
604     System.out.println();
605     System.out.println("List of available time-slots:");
606     System.out.println();
607     System.out.println("Please, enter one of the following:");
608     System.out.println();
609
610     //4 lists to search for available time-slots
611     ArrayList<AssistantOnShift> listAssistant = new ArrayList<AssistantOnShift>();
612     ArrayList<BookableRoom> listBookableRooms = new ArrayList<BookableRoom>();
613
614     ArrayList<AssistantOnShift> listAssistantFinal = new ArrayList<AssistantOnShift>();
615     ArrayList<BookableRoom> listBookableRoomsFinal = new ArrayList<BookableRoom>();
616
617     //find all bookable rooms which are empty or available
618     for (BookableRoom b : BookingSystem.getBookableRooms()){
619         String statusBooking = b.findOutStatusRoom();
620         if (statusBooking == "EMPTY" || statusBooking == "AVAILABLE"){
621             listBookableRooms.add(b);
622         }
623     }
624
625     //find all assistatns which are free
626     for (AssistantOnShift a : BookingSystem.getAssistantOnShifts()){
627         String stasisAssistant = a.findOutStatusAssistan();
628         if (stasisAssistant == "FREE"){
629             listAssistant.add(a);
630         }
631     }
632
633     //look where both lists have same dates
634     //meaning finding free time spaces
635     int indx = 1;
636     for (BookableRoom b : listBookableRooms){
637         int c = b.carryCampacity();
638         for (int i = 0; i <= c; i++){
639             for (AssistantOnShift a : listAssistant){
640                 if (b.getDate().equals(a.getDate())){
641                     System.out.println(indx + ". " + a.getDate() + " " + b.getTime());
642                     listAssistantFinal.add(a);
643                     listBookableRoomsFinal.add(b);
644                     indx = indx + 1;
645                 }
646             }
647         }
648     }
649
650     //rest format of the page
651     System.out.println("The sequential ID of an available time-slot and the student email, separated by a white space.");
652     System.out.println("0. Back to main menu.");
653     System.out.println("-1. Quit application.");
654     System.out.println();
655
656     //flag for continious calls
657     boolean nw_flg = true;

```

```
boolean nw_flag = true;
```

```
//loop with the flag
```

```
while (nw_flag == true) {
```

```
    //getting user input
```

```
    Scanner sc = new Scanner(System.in);
```

```
    String[] us_in = sc.nextLine().split(" ");
```

```
    //exit to main menu
```

```
    if (us_in[0].equals("0")){
```

```
        nw_flag = false;
```

```
    //exit the system
```

```
    } else if (us_in[0].equals("-1")){
```

```
        System.exit(0);
```

```
    }
```

```
    //try adding the booking
```

```
    try {
```

```
        //if statment to control user input
```

```
        if (nw_flag == true) {
```

```
            //put user input to correct format
```

```
            int i = Integer.parseInt(us_in[0]);
```

```
            String studentEmail = us_in[1];
```

```
            //validate the student email
```

```
            BookingSystem.validateStudentEmail(studentEmail);
```

```
            //get the correct assistant on shift and bookable room
```

```
            BookableRoom br = listBookableRoomsFinal.get(i-1);
```

```
            AssistantOnShift as = listAssistantFinal.get(i-1);
```

```
            //create the booking
```

```
            BookingSystem.addBooking(as, br, br.getTime(), studentEmail, "SCHEDULED");
```

```
            System.out.println("Booking added successfully:");
```

```
            //print that booking
```

```
            System.out.println(BookingSystem.getBookings().get(BookingSystem.getBookings().size()-1).printThe());
```

```
            System.out.println();
```

```
            System.out.println("Please, enter one of the above!");
```

```
            System.out.println();
```

```
            System.out.println("The sequential ID of an available time-slot and the student email, separated by a white space.");
```

```
            System.out.println("0. Back to main menu.");
```

```
            System.out.println("-1. Quit application.");
```

```
            System.out.println();
```

```
        } else {
```

```
            System.out.println();
```

```
        }
```

```
    //if error occures
```

```
    } catch (Exception e) {
```

```
        //print the fail form
```

```
        System.out.println("Error!");
```

```
        System.out.println("This input is invaild, please follow the correct syntax");
```

```
        System.out.println("Please, enter one of the above!");
```

```
        System.out.println();
```

```
        System.out.println("The sequential ID of an available time-slot and the student email, separated by a white space.");
```

```
        System.out.println("0. Back to main menu.");
```

```
        System.out.println("-1. Quit application.");
```

```
        System.out.println();
```

```
    }
```

```
}
```

```
}
```

```
//ninth page
```

```
if (user_input.equals("9")){
```

```
    //format of the page
```

```
    System.out.println("University of Knowledge - COVID test");
```

```
    System.out.println();
```

```
    System.out.println("Removing booking from the system");
```

```
    System.out.println();
```

```

731 System.out.println();
732 System.out.println("Please, enter one of the following:");
733 System.out.println();
734
735 //print all bookings that can be removed
736 for (Booking b: BookingSystem.getBookings()){
737     String status = b.findOutStatusBooking();
738     if (status == "SCHEDULED"){
739         int k = BookingSystem.getBookings().indexOf(b);
740         System.out.print(k + 1 + ". ");
741         System.out.println(b.printThe());
742         k = k + 1;
743     }
744 }
745
746 //rest format of the page
747 System.out.println("The sequential ID to select the booking to be removed from the listed bookings above.");
748 System.out.println("0. Back to main menu.");
749 System.out.println("-1. Quit application.");
750 System.out.println();
751
752 //flag for continious calls
753 boolean new_new_flg = true;
754
755 //loop with the flag
756 while (new_new_flg == true){
757
758     //getting user input
759     Scanner sc = new Scanner(System.in);
760     String us_in = sc.nextLine();
761
762     //exit to main menu
763     if (us_in.equals("0")){
764         new_new_flg = false;
765     }
766
767     //exit the system
768     } else if (us_in.equals("-1")){
769         System.exit(0);
770     }
771
772     //try removing the booking
773     try{
774         //if statment ot control user input
775         if (new_new_flg == true){
776             //convert user input to integer
777             int i = Integer.parseInt(us_in);
778
779             //print the booking which will be removed
780             System.out.println("Booking removed successfully.");
781             System.out.println(BookingSystem.getBookings().get(i-1).printThe());
782
783             //remove booking
784             BookingSystem.removeBooking(BookingSystem.getBookings().get(i-1));
785
786             //rest of the format
787             System.out.println("Please, enter one of the above!");
788             System.out.println();
789             System.out.println("The sequential ID to select the booking to be removed from the listed bookings above.");
790             System.out.println("0. Back to main menu.");
791             System.out.println("-1. Quit application.");
792             System.out.println();
793         } else{
794             System.out.println();
795         }
796     } //if error occures
797     } catch (Exception e){
798
799         //print the fail form
800         System.out.println("Error!");
801         System.out.println("This input is invaild, please follow the correct syntax");
802         System.out.println();
803         System.out.println("Please, enter one of the above!");
804         System.out.println();
805         System.out.println("The sequential ID to select the booking to be removed from the listed bookings above.");
806         System.out.println("0. Back to main menu.");

```

```
805     System.out.println("0. Back to main menu. ");
806     System.out.println("-1. Quit application.");
807     System.out.println();
808
```

```
809 }
810 }
811 }
```

```
812
813 //tenth page
```

```
814 if (user_input.equals("10")){
```

```
815
816     //format of the page
```

```
817     System.out.println("University of Knowledge - COVID test");
```

```
818     System.out.println();
```

```
819     System.out.println("Conclude booking");
```

```
820     System.out.println();
```

```
821     System.out.println("Please, enter one of the following:");
```

```
822     System.out.println();
823
```

```
824     //print the bookings which can be concluded
```

```
825     for (Booking b: BookingSystem.getBookings()){
```

```
826         String status = b.findOutStatusBooking();
```

```
827         int k = BookingSystem.getBookings().indexOf(b);
```

```
828         if (status == "SCHEDULED"){
```

```
829             System.out.print(k+1 + ". ");
```

```
830             System.out.println(b.printThe());
```

```
831             k = k + 1;
```

```
832         }
```

```
833     }
```

```
834
835     //rest format of the page
```

```
836     System.out.println("The sequential ID to select the booking to be completed.");
```

```
837     System.out.println("0. Back to main menu.");
```

```
838     System.out.println("-1. Quit application.");
```

```
839     System.out.println();
840
```

```
841     //flag for continious calls
```

```
842     boolean new_new_flg = true;
```

```
843
844     //loop with the flag
```

```
845     while (new_new_flg == true){
```

```
846
847         //getting user input
```

```
848         Scanner sc = new Scanner(System.in);
```

```
849         String us_in = sc.nextLine();
850
```

```
851         //exit to main menu
```

```
852         if (us_in.equals("0")){
```

```
853             new_new_flg = false;
```

```
854
855         //exit the system
```

```
856     } else if (us_in.equals("-1")){
```

```
857         System.exit(0);
```

```
858     }
```

```
859
860     //try to conclude booking
```

```
861     try{
```

```
862         //if statment to control user input
```

```
863         if (new_new_flg == true){
```

```
864
865             //convert user input to integer
```

```
866             int i = Integer.parseInt(us_in);
```

```
867             //change status to completed
```

```
868             BookingSystem.getBookings().get(i-1).setStatus("COMPLETED");
```

```
869             //print the form and the booking which is now concluded
```

```
870             System.out.println("Booking completed successfully.");
```

```
871             System.out.println(BookingSystem.getBookings().get(i-1).printThe());
```

```
872             System.out.println("Please, enter one of the above!");
```

```
873             System.out.println();
```

```
874             System.out.println("The sequential ID to select the booking to be completed.");
```

```
875             System.out.println("0. Back to main menu.");
```

```
876             System.out.println("-1. Quit application.");
```

```
877             System.out.println();
878
```

```

879     }else{
880         System.out.println();
881     }
882
883     //if error occurs
884     } catch (Exception e){
885         //print the fail form
886         System.out.println("Error!");
887         System.out.println("This input is invaild, please follow the correct syntax");
888         System.out.println("Please, enter one of the above!");
889         System.out.println();
890         System.out.println("The sequential ID to select the booking to be completed.");
891         System.out.println("0. Back to main menu.");
892         System.out.println("-1. Quit application.");
893         System.out.println();
894     }
895 }
896 }
897
898 //way to exit from the main menu
899 if (flag == false)
900     break;
901 }
902
903 }
904 }

```