

Table of Contents

<u>1. Unit Test Results</u>	1
Functionalities Tested	2
Unit TestCases Executed	2
<u>2. Functionality and Paths</u>	3
<u>authenticate: Healthcare App</u>	3
View Appointments: HealthcareApp	5
Make appointment: HealthcareApp	7
Search user: Healthcare App	9
View Reports: HealthcareApp	14
Submit Report: HealthcareApp	19

1. Unit Test Results

Functionality Name	Class Name	Pass /Fail	Defects	Tester
<u>authenticate: Healthcare App</u>	HealthcareApp	pass	0	Bharadwaj
<u>View Appointments: HealthcareApp</u>	HealthcareApp	Pass	0	Bharadwaj
<u>Make appointment: HealthcareApp</u>	HealthcareApp	pass	0	Bharadwaj
<u>Search user: Healthcare App</u>	HealthcareApp	Pass	0	Bharadwaj
<u>View Reports: HealthcareApp</u>	HealthcareApp	pass	0	Bharadwaj
<u>Submit Report: HealthcareApp</u>	HealthcareApp	Pass	0	Bharadwaj

1.1. Functionalities Tested

1.2. Unit TestCases Executed

Unit Test Case ID	Unit test case name	Pass	Defect count	Tested By
1	testAuthenticate	Pass	0	Bharadwaj
2	testAuthenticateFalse	Pass	0	Bharadwaj
3	testViewAppointments	Pass	0	Bharadwaj
4	testViewAppointmentsFalse	Pass	0	Bharadwaj
5	testMakeAppointments	Pass	0	Bharadwaj
6	testMakeAppointmentsFalse	Pass	0	Bharadwaj
7	testSearchuser	Pass	0	Bharadwaj
8	testSearchuserFalse	Pass	0	Bharadwaj
9	testViewReport	Pass	0	Bharadwaj
10	testViewReportFalse	Pass	0	Bharadwaj
11	testSubmitReport	Pass	0	Bharadwaj
12	testSubmitReportFalse	Pass	0	Bharadwaj
13	testisPatient	Pass	0	Bharadwaj
14	testisDoctor	Pass	0	Bharadwaj
15	testGetType	Pass	0	Bharadwaj
16	testGetPassHash	Pass	0	Bharadwaj
17	testGetUsername	Pass	0	Bharadwaj
18	testGetLastLogin	Pass	0	Bharadwaj

The image displays two screenshots of an IDE's test runner window, showing the results of unit tests for a healthcare application.

Top Screenshot: healthcareapp.HealthcareAppIT

- 100.00 %
- All 11 tests passed. (1.438 s)
- healthcareapp.HealthcareAppIT passed
 - testSubmitReportFalse passed (0.858 s)
 - testMain passed (0.0 s)
 - testAuthenticate passed (0.047 s)
 - testHash passed (0.0 s)
 - testViewReports passed (0.095 s)
 - testSearchUser passed (0.021 s)
 - testSubmitReport passed (0.077 s)
 - testViewAppointments passed (0.048 s)
 - testViewAppointmentsFalse passed (0.044 s)
 - testMakeAppointment passed (0.044 s)
 - testAuthenticateFalse passed (0.035 s)

Right Panel (Top Screenshot):

```
doctor
2015-07-15, with Doctor Clinton Jarboe
2015-03-27, with Doctor Clinton Jarboe
2015-03-25, with Doctor Clinton Jarboe
2015-04-15, with Doctor Jay Clinton
viewAppointments
cjarbo2 ichabod doctor 2015-03-15
cjarbo2, ichabod, 1, 2015-03-15
cjarbo2
doctor
doctor
2015-07-15, with patient Jay Clinton
2015-03-27, with patient Jay Clinton
2015-03-25, with patient Jay Clinton
cdjarboe ichabod2 patient 2015-04-17
cdjarboe, ichabod2, 0, 2015-04-17
Enter the patient's full name:
authenticate
Invalid username.
cjarbo2 ichabod doctor 2015-03-15
cjarbo2, ichabod, 1, 2015-03-15
```

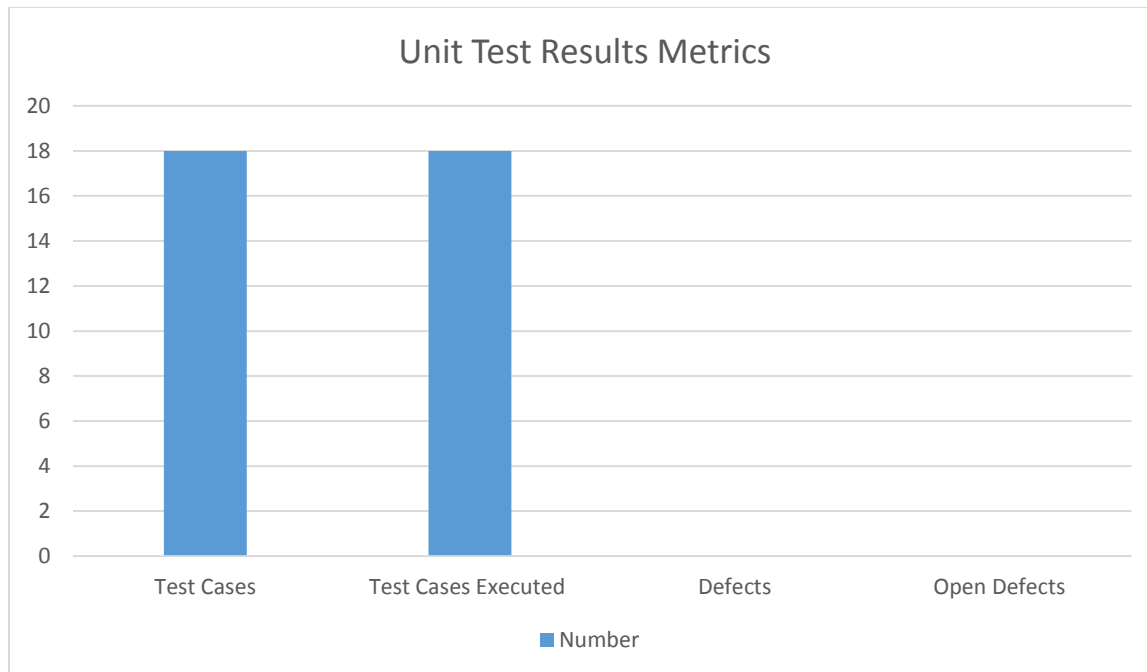
Bottom Screenshot: healthcareapp.UserIT

- 100.00 %
- All 6 tests passed. (0.196 s)
- healthcareapp.UserIT passed
 - testIsDoctor passed (0.004 s)
 - testGetType passed (0.001 s)
 - testIsPatient passed (0.001 s)
 - testGetPassHash passed (0.0 s)
 - testGetUserName passed (0.0 s)
 - testGetLastLogin passed (0.0 s)

Right Panel (Bottom Screenshot):

```
dummy doctor Wed Dec 31 19:48:22 MST 1969
dummy patient Wed Dec 31 19:48:22 MST 1969
or

dummy doctor Wed Dec 31 19:48:22 MST 1969
dummy patient Wed Dec 31 19:48:22 MST 1969
e
dummy doctor Wed Dec 31 19:48:22 MST 1969
dummy patient Wed Dec 31 19:48:22 MST 1969
ent
dummy doctor Wed Dec 31 19:48:22 MST 1969
dummy patient Wed Dec 31 19:48:22 MST 1969
sHash
dummy doctor Wed Dec 31 19:48:22 MST 1969
dummy patient Wed Dec 31 19:48:22 MST 1969
rName
dummy doctor Wed Dec 31 19:48:22 MST 1969
dummy patient Wed Dec 31 19:48:22 MST 1969
tLogin
```



2 Functionality and Paths

1. authenticate: Healthcare App

Precondition: User has been registered and has application available.

Post-condition: User is logged into Application

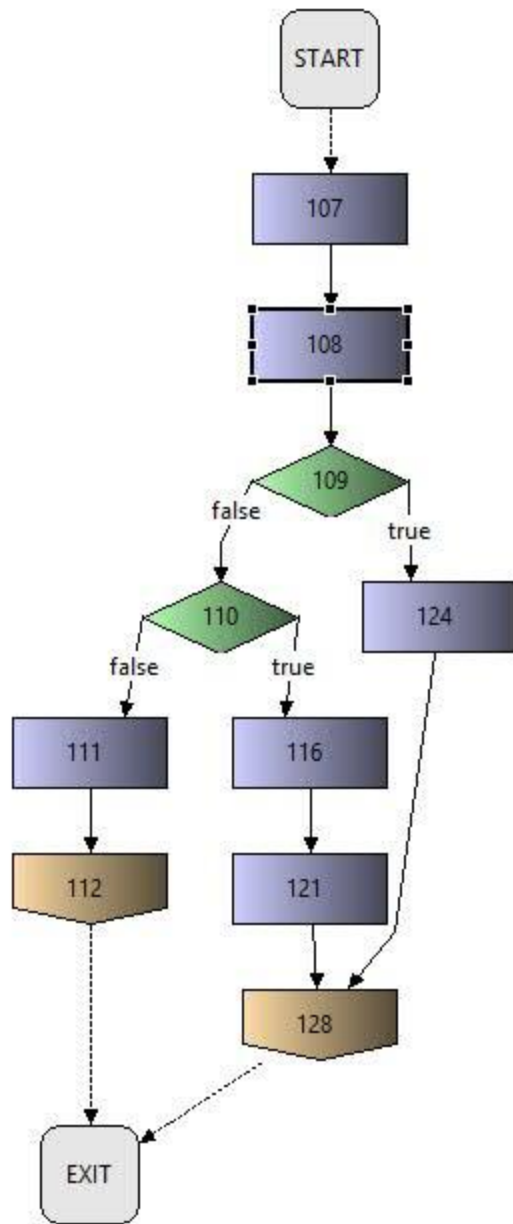
Behavior: The method takes user username and password as input and compares with database to check if it correct and authenticate it.

```
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131

public static boolean authenticate(String userName, String passHash) {

    User found = searchUser(userName);
    passHash = hash(passHash);
    if (found != null) {
        if (passHash.equals(found.getPassHash())) {
            userprofile = found;
            return true;
        }
        else {
            System.out.println("Invalid password.");
        }
    }
    else {
        System.out.println("Invalid username.");
    }

    return false;
}
```



NOTE: the numbers in blocks of the graph are line numbers on the code and haven't been changed.

Number of independent paths: $2+1=3$

PATHS:

Path 1: 107-108-109-124-128 Test Case (username is incorrect)

Path 2: 107-108-109-110-111-112 Test Case (Username and password is incorrect)

Path 3: 107-108-109-110-116-121-128 Test Case (Password is incorrect)

2. View Appointments: HealthcareApp

Precondition: Login is successful. Has at least one appointment

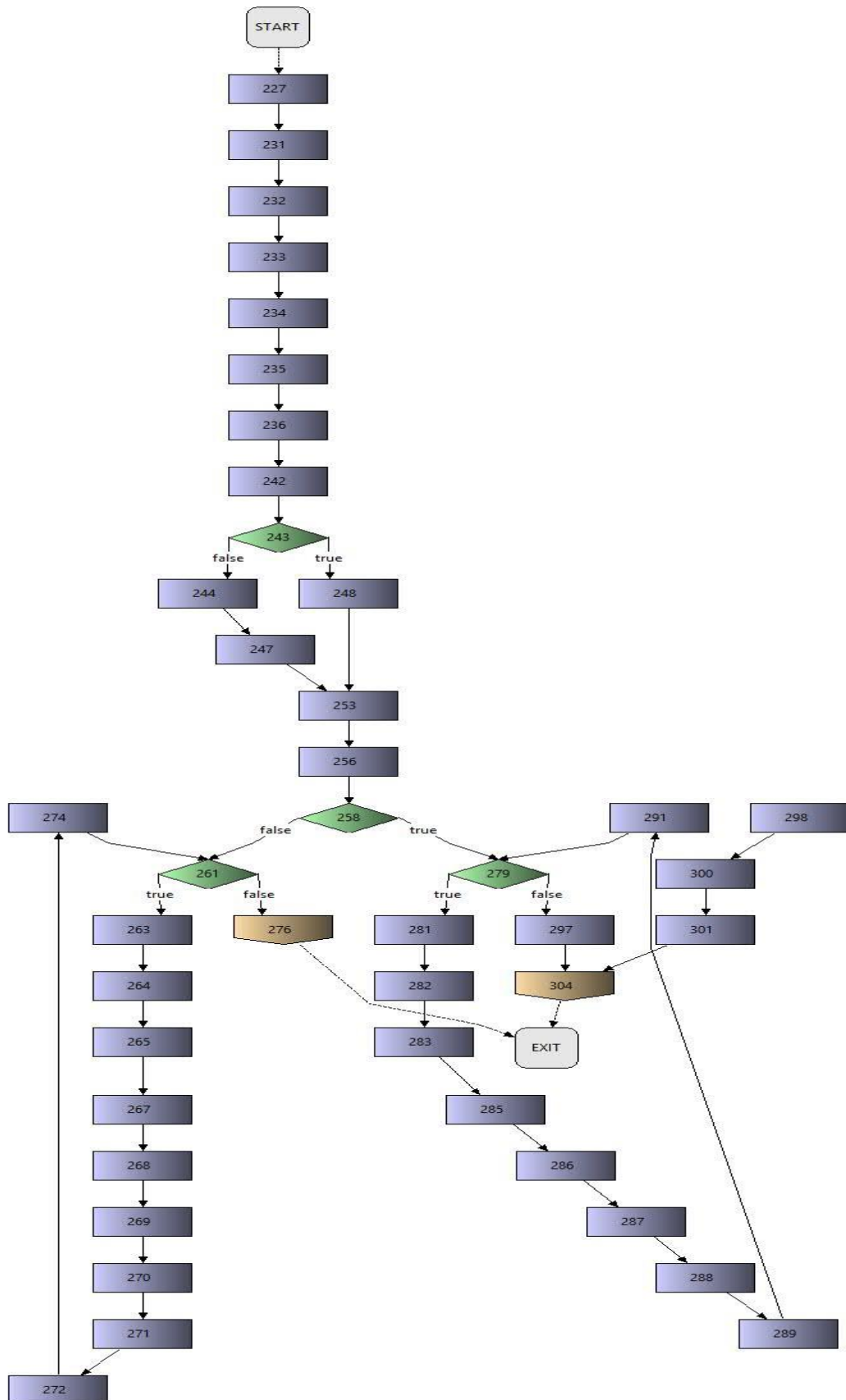
Post condition: Able to appointments.

Behavior: After a user is logged in based on the user shows the appropriate appointments, time, date, doctor, patient and other details. A patient can see his appointments with Doctor in descending order.

A Doctor can see his appointment with patients.

```
226 ☐ public static String viewAppointments() {
227     String b = null;
228     try
229     {
230         // create our mysql database connection
231         String myDriver = "org.gjt.mm.mysql.Driver";
232         String myUrl = URL;
233         String user = USERNAME;
234         String password = PASSWORD;
235         Class.forName(myDriver);
236         Connection conn = DriverManager.getConnection(myUrl, user, password);
237
238         // our SQL SELECT query.
239         // if you only need a few columns, specify them by name instead of using "*"
240         String query;
241         // System.out.println(userprofile.isPatient());
242         System.out.println(userprofile.getUserName());
243         if (userprofile.isPatient()) {
244             query = "SELECT * FROM appointments WHERE patient='"+userprofile.getUserName()+"'";
245
246             //System.out.println("Test...PatientOK");
247         }
248         else query = "SELECT * FROM appointments WHERE doctor='"+userprofile.getUserName()+"'";
249
250         // System.out.println(query);
251
252         // create the java statement
253         Statement st = conn.createStatement();
```

```
254
255 // execute the query, and get a java resultset
256 ResultSet rs = st.executeQuery(query);
257
258 if (userprofile.isDoctor()) {
259
260     // iterate through the java resultset
261     while (rs.next())
262     {
263         b="doctor";
264         Date date = rs.getDate("date");
265         String patKey = rs.getString("patient");
266         // doctor = SELECT fullname FROM users WHERE userName='docKey'
267         query = "SELECT fullname FROM users WHERE userName='"+patKey+"'";
268         Statement st2 = conn.createStatement();
269         ResultSet rs2 = st2.executeQuery(query);
270         rs2.next();
271         String patient = rs2.getString("fullname");
272         st2.close();
273         // print the results
274         System.out.format("%s, with patient %s\n", date, patient);
275     }
276     return("Doctor");
277 }
278 else {
279     while (rs.next())
280     {
281         b="patient";
282         Date date = rs.getDate("date");
283         String docKey = rs.getString("doctor");
284         // doctor = SELECT fullname FROM users WHERE userName='docKey'
285         query = "SELECT fullname FROM users WHERE userName='"+docKey+"'";
286         Statement st2 = conn.createStatement();
287         ResultSet rs2 = st2.executeQuery(query);
288         rs2.next();
289         String doctor = rs2.getString("fullname");
290         // print the results
291         System.out.format("%s, with Doctor %s\n", date, doctor);
292     }
293
294 }
295
296 //st.close();
297
298 }
299 catch (Exception e)
300 {
301     System.err.println("Got an exception! ");
302     System.err.println(e.getMessage());
303 }
304
305 return(b);
}
```

Path 1: 227-231-232-233-234-235-236-242-243-248-253-256-258-279-297-304 Test case (patient)

Path 2: 227-231-232-233-234-235-236-242-243-248-253-256-258-279-281-282-283-285-286-287-288-289-291-279-297-304 Test Case Doctor

Path 3: 227-231-232-233-234-235-236-242-243-248-253-256-258-261-276 Test Case Doctor

Path 4: 227-231-232-233-234-235-236-242-243-248-253-256-258-261-276

Path 5: 227-231-232-233-234-235-236-242-243-248-253-256-258-261-263-264-265-267-268-269-270-271-274-261-276 Test Case Doctor

3. Make appointment: HealthcareApp

Pre-Condition: User is able to login and has permission to make appointment.

Post-condition: Able to select a doctor and make appointment.

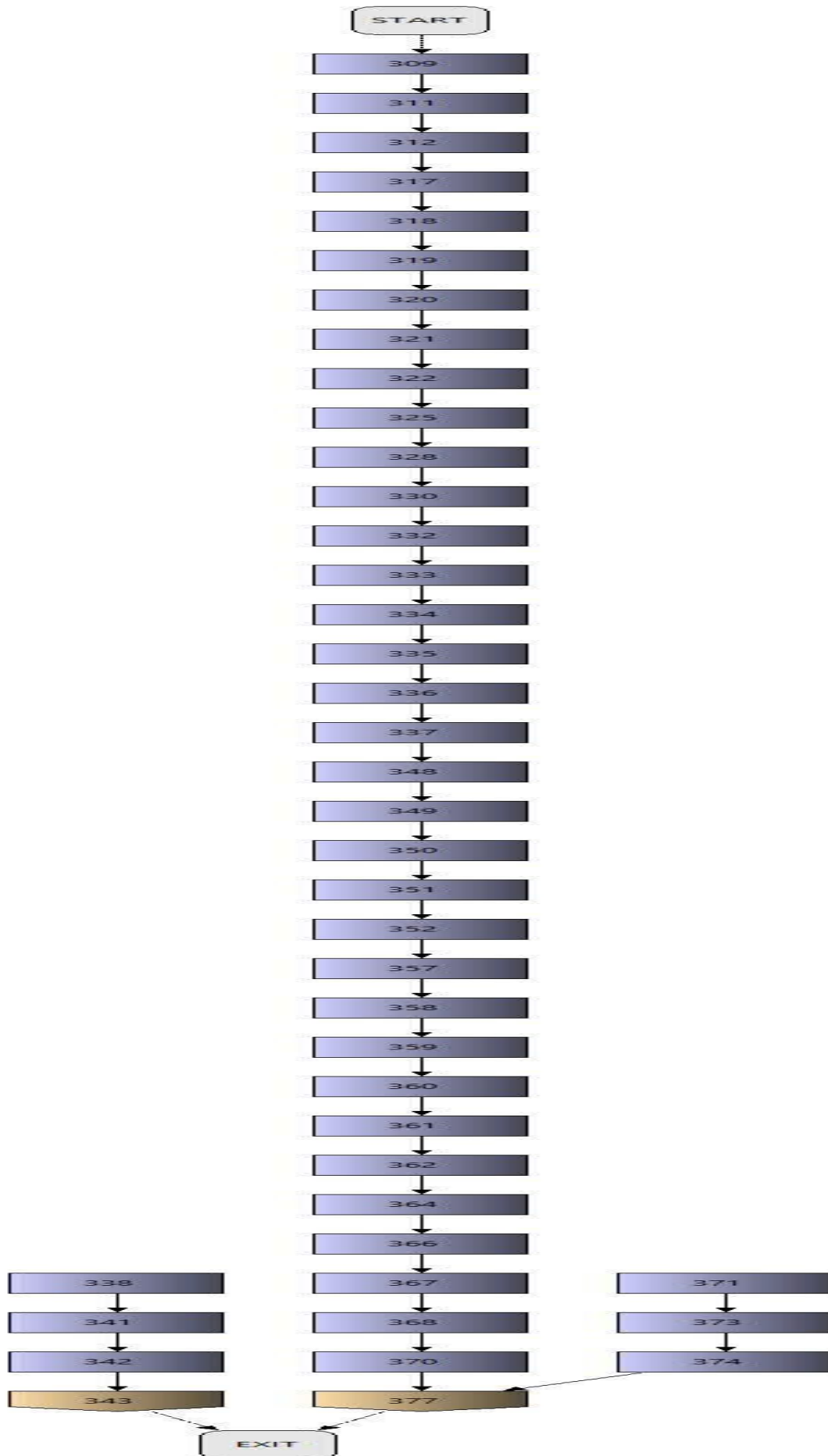
Behavior: Make Appointment helps to schedule appointment for a patient by healthcare professional

User Enters Patient Id, Doctor, time, date and Appointment is set at the given time.

Only authorized Personal can set Appointment.

```
307 public static boolean makeAppointment () {
308
309     Scanner reader = new Scanner(System.in);
310
311     System.out.println("Enter the patient's full name: ");
312     String patient = reader.nextLine();
313
314     try
315     {
316         // create mysql database connection
317         String myDriver = "org.gjt.mm.mysql.Driver";
318         String myUrl = URL;
319         String user = USERNAME;
320         String password = PASSWORD;
321         Class.forName(myDriver);
322         Connection conn = DriverManager.getConnection(myUrl, user, password);
323
324         //DateFormat df = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss");
325         Date date = new Date();
326
327         //String date = df.format(d);
328         SimpleDateFormat dt = new SimpleDateFormat("yyyyMMddhhmmss");
329
330         String query = "SELECT userName FROM users WHERE fullname='"+patient+"'";
331
332         Statement st = conn.createStatement();
333         ResultSet rs = st.executeQuery(query);
334         rs.next();
```

```
335         patient = rs.getString("userName");
336         st.close();
337     }
338     catch (Exception e)
339     {
340
341         System.err.println("Got an exception! ");
342         System.err.println(e.getMessage());
343         return false;
344
345
346     }
347
348     System.out.println("Enter the time of the appointment, in the form hhmmss : ");
349     String time = reader.nextLine();
350     System.out.println("Enter the date of the appointment, in the form yyyyMMdd : ");
351     String date = reader.nextLine();
352     date = date + time;
353
354     try
355     {
356         // create mysql database connection
357         String myDriver = "org.gjt.mm.mysql.Driver";
358         String myUrl = URL;
359         String user = USERNAME;
360         String password = PASSWORD;
361         Class.forName(myDriver);
362         Connection conn = DriverManager.getConnection(myUrl, user, password);
363
364         String query = "INSERT INTO appointments (patient, doctor, date) "
365             + "VALUES ('"+patient+"', '"+userprofile.getUserName()+"', '"+date+"')";
366
367         Statement st = conn.createStatement();
368         st.executeUpdate(query);
369         st.close();
370
371     }
372     catch (Exception e)
373     {
374         System.err.println("Got an exception! ");
375         System.err.println(e.getMessage());
376
377     }
378     return false;
379
380 }
```



Path 1: 309-310-311-312-317-319-320-322-325-328-330-332-333-334-335-336-337-348-349-350-351-357-358-359-360-361-362-364-366-367-368-370-372

4. Search user: Healthcare App

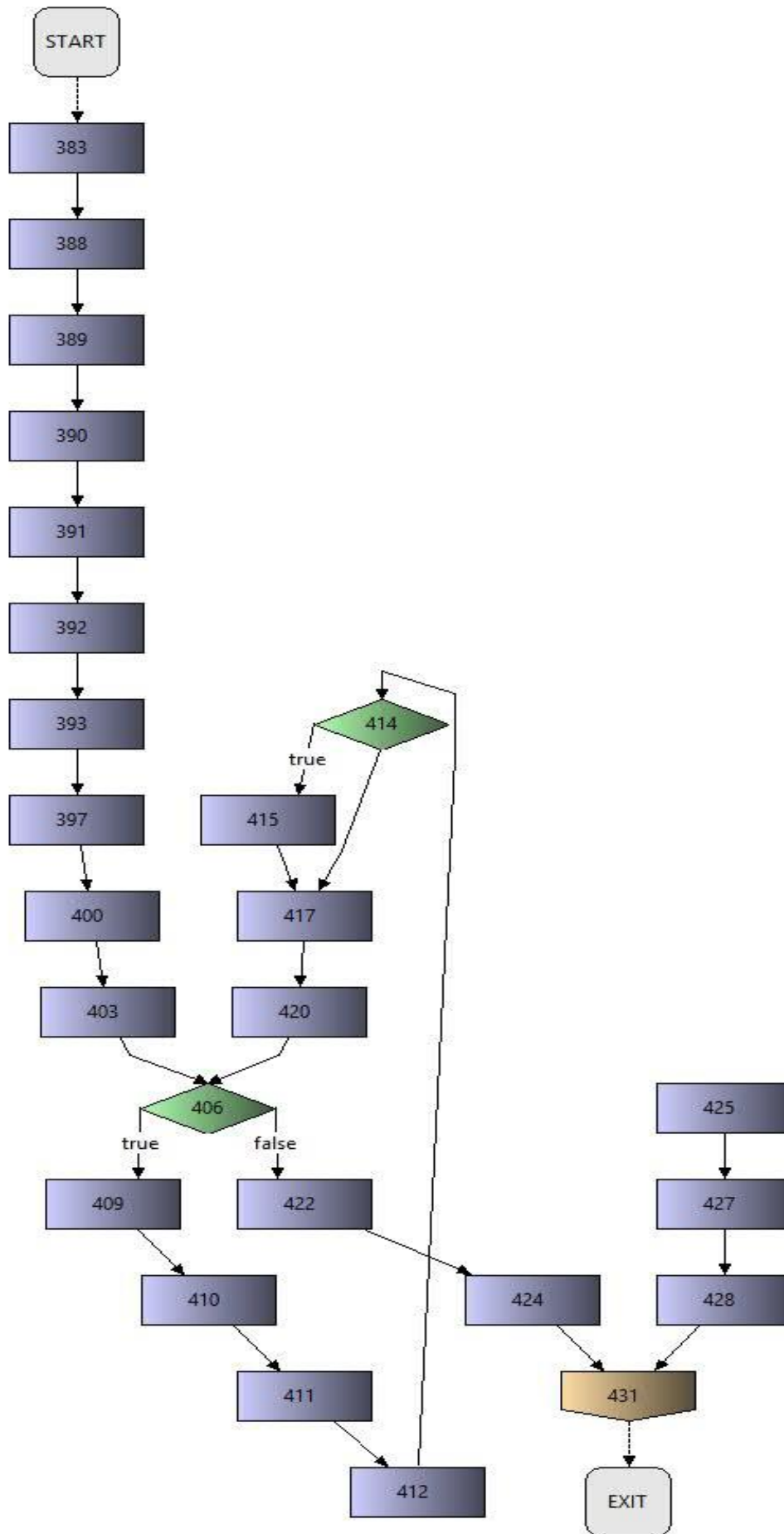
Precondition: User is Log in to the System.

Post Condition: None.

Behavior: Searches in the database if the User is available or not.

```
382 public static User searchUser(String userName) {
383
384     User found = null;
385
386     try
387     {
388         // create our mysql database connection
389         String myDriver = "org.gjt.mm.mysql.Driver";
390         String myUrl = URL;
391         String user = USERNAME;
392         String password = PASSWORD;
393         Class.forName(myDriver);
394         Connection conn = DriverManager.getConnection(myUrl, user, password);
395
396         // our SQL SELECT query.
397         // if you only need a few columns, specify them by name instead of using "*"
398         String query = "SELECT * FROM users WHERE userName='"+userName+"'";
399
400         // create the java statement
401         Statement st = conn.createStatement();
402
403         // execute the query, and get a java resultset
404         ResultSet rs = st.executeQuery(query);
405
406         // iterate through the java resultset
407         while (rs.next())
408         {
409
410             String qUserName = rs.getString("userName");
```

```
411         String qPassHash = rs.getString("passHash");
412         int usertype = rs.getInt("userType");
413         Date lastlogin = rs.getDate("lastLogin");
414         userType uType;
415         if (usertype == 0) uType = userType.patient;
416         else uType = userType.doctor;
417
418         found = new User(qUserName, qPassHash, uType, lastlogin);
419
420         // print the results
421         System.out.format("%s, %s, %s, %s\n", qUserName, qPassHash, usertype, lastlogin);
422     }
423     st.close();
424
425 }
426
427 catch (Exception e)
428 {
429     System.err.println("Got an exception! ");
430     System.err.println(e.getMessage());
431 }
432
433 return found;
434 }
```



Paths: 2+1=3

Path 1:383-388-389-390-391-392-393-397-400-403-406-424-431

Path 2: 383-388-389-390-391-392-393-397-400-403-406-409-410-411-412-414-415-417-420-406-422-424-431

Path 3: 383-388-389-390-391-392-393-397-400-403-406-409-410-411-412-414-415-417-420-406-422-424-431

5. View Reports: HealthcareApp

Pre-Condition: User is logged in and has reports created.

Post-Condition: None

Behavior: User after log in on home page is able to see a view reports button. Selecting a report on the reports displayed and clicking view report button shows reports.

Reports contain doctor name, symptoms of patient and comments.

```
436 public static String viewReports() {
437     String b="none";
438     try
439     {
440         // create our mysql database connection
441         String myDriver = "org.gjt.mm.mysql.Driver";
442         String myUrl = URL;
443         String user = USERNAME;
444         String password = PASSWORD;
445         Class.forName(myDriver);
446         Connection conn = DriverManager.getConnection(myUrl, user, password);
447         Connection conn2 = DriverManager.getConnection(myUrl, user, password);
448
449         String query;
450
451         Statement st = conn.createStatement();
452         Statement st2 = conn2.createStatement();
453         ResultSet rs;
454
455         Date date;
456         int pain, drowsiness, nausea, anxiety, depression;
457         String comments;
458         if (userprofile.isPatient()) {
459             query = "SELECT * FROM reports WHERE patient='"+userprofile.getUserName()+"' ORDER BY dateTime ASC";
460             System.out.println(userprofile.getUserName());
461             //System.out.println("Test...PatientOK");
462             rs = st.executeQuery(query);
```



```

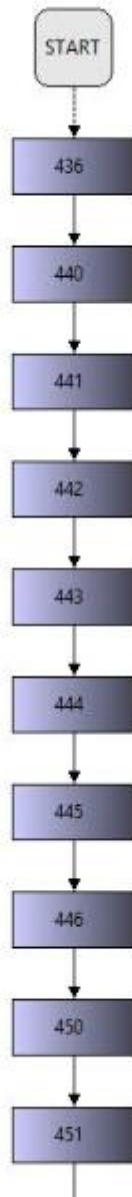
464         {
465             date = rs.getDate("dateTime");
466             pain = rs.getInt("pain");
467             drowsiness = rs.getInt("drowsiness");
468             nausea = rs.getInt("nausea");
469             anxiety = rs.getInt("anxiety");
470             depression = rs.getInt("depression");
471             comments = rs.getString("comments");
472             System.out.format("%s, pain: %d, drowsiness: %d, nausea: %d, anxiety: %d, depression: %d, Comments: %s",
473                             patientName, pain, drowsiness, nausea, anxiety, depression, comments);
474         }
475         b="patient";
476     }
477     else {
478         int i = 0;
479         int j;
480         query = "SELECT * FROM patientdoctorjoin WHERE doctor='"+userprofile.getUserName()+"'";
481         //System.out.println(query);
482         rs = st.executeQuery(query);
483         ResultSet rs2;
484         String patient, patientName;
485         System.out.println("Before first while");
486         while (rs.next()) {
487             i++;
488             System.out.println("After "+i+" instance of first while");
489             patient = rs.getString("patient");
490             query = "SELECT fullname FROM users WHERE userName='"+patient+"'";
491             //System.out.println(query);
492             rs2 = st2.executeQuery(query);
493             rs2.next();
494             patientName = rs2.getString("fullname");
495             query = "SELECT * FROM reports WHERE patient='"+patient+"' ORDER BY dateTime ASC";
496             //System.out.println(query);
497             rs2 = st2.executeQuery(query);
498             System.out.println("Before 2nd while");
499             j=0;
500             while (rs2.next()) {
501                 j++;
502                 System.out.println("After 2nd while instance "+j);
503                 date = rs2.getDate("dateTime");
504                 pain = rs2.getInt("pain");
505                 drowsiness = rs2.getInt("drowsiness");
506                 nausea = rs2.getInt("nausea");
507                 anxiety = rs2.getInt("anxiety");
508                 depression = rs2.getInt("depression");
509                 comments = rs2.getString("comments");
510                 System.out.format("%s, %s, pain: %d, drowsiness: %d, nausea: %d, anxiety: %d, depression: %d, Comments: %s",
511                                 patientName, patient, date, pain, drowsiness, nausea, anxiety, depression, comments);
512             }
513             rs2.close();
514             st2.close();
515         }
516         b="doctor";
517     }
518     rs.close();
519     st.close();
520     return(b);
521 }

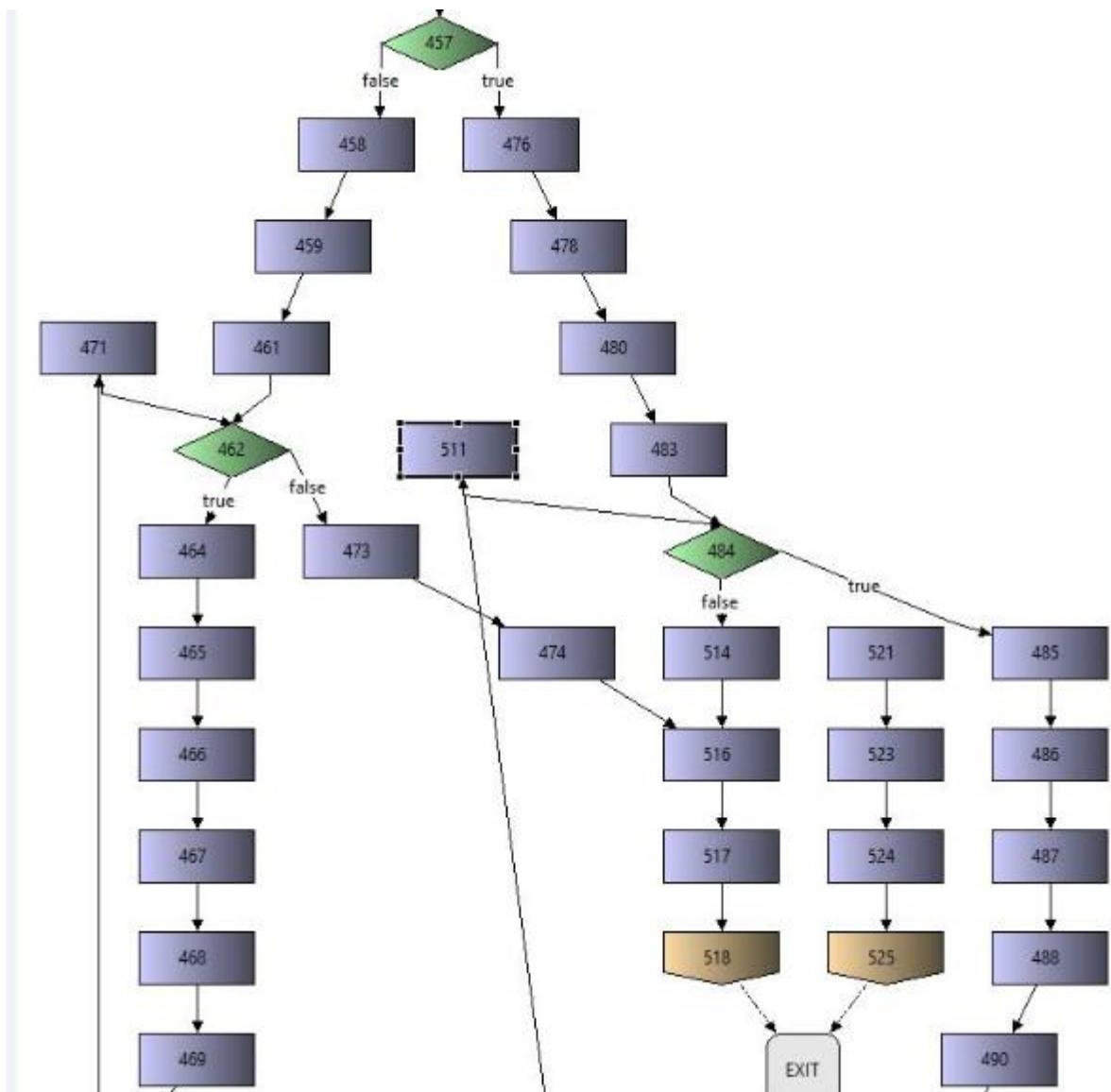
```

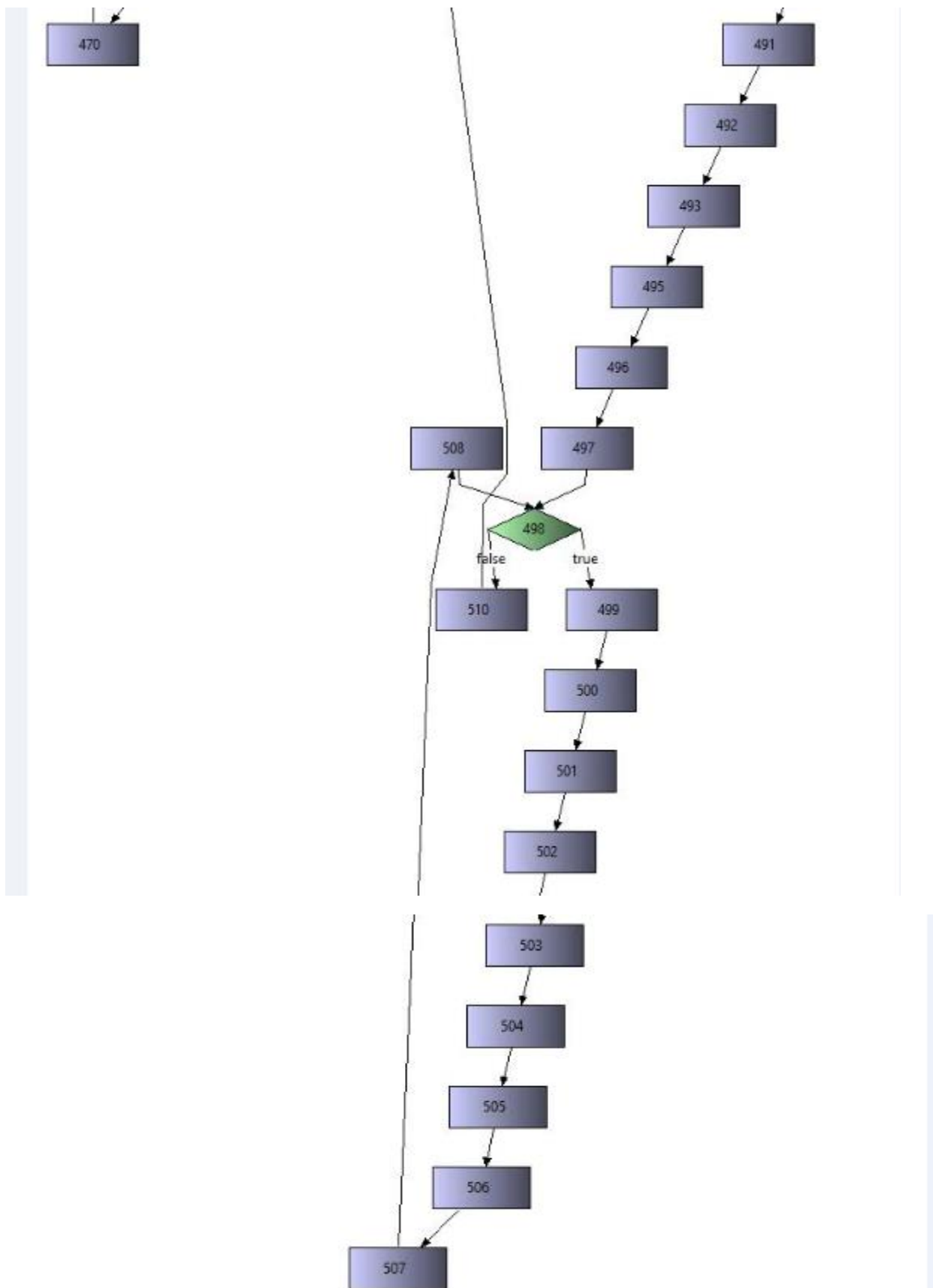
```

523     catch (Exception e)
524     {
525         System.err.println("Got an exception! ");
526         System.err.println(e.getMessage());
527         return("none");
528     }
529 }
530

```







Paths 4+1=5

Path1:436+440+441+442+443+444+445+446+450+451+457true+476+478+480+483+484true+485+486+487+488+490+491+492+493+495+496+497+498true+499+500+504+507+508+510+511+484false+514+516+517+518

Path2:436+440+441+442+443+444+445+446+450+451+457true+476+478+480+483+484false+514+516+517+518

Path3:436+440+441+442+443+444+445+446+450+451+457false+458+459+461+462true+464+466+469+471+462false+473+474+516+517+518

Path4:436+440+441+442+443+444+445+446+450+451+457false 473+474+516+517+518

Path5:436+440+441+442+443+444+445+446+450+451+457+476+478+480+483+484true+485+486+487+488+490+491+492+493+495+496+497+498false+510+511+484false+514+516+517+518

6. Submit Report: HealthcareApp

Precondition: User is logged in and has authorization to submit report.

Post-condition: None

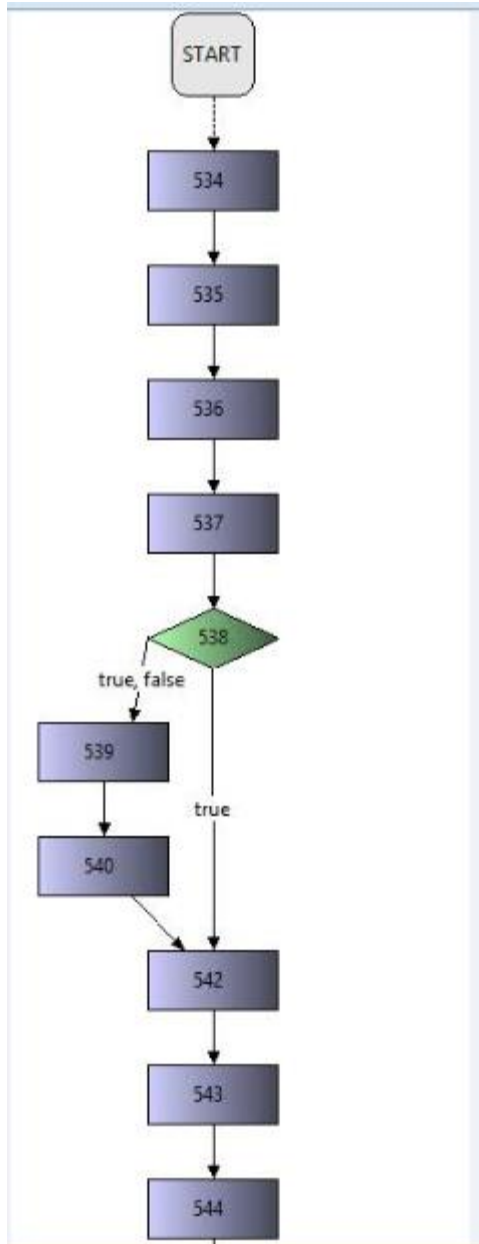
Behavior:

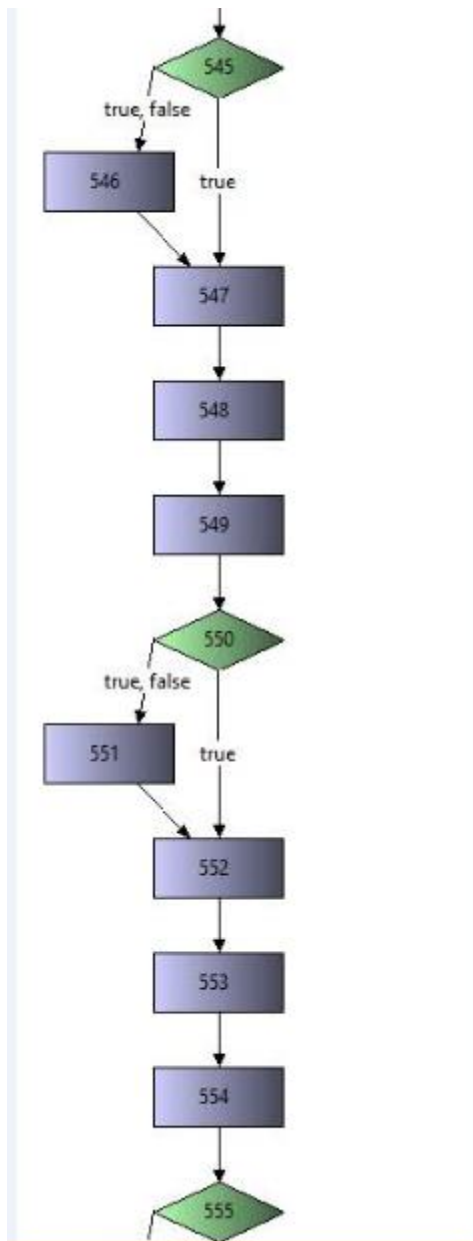
```

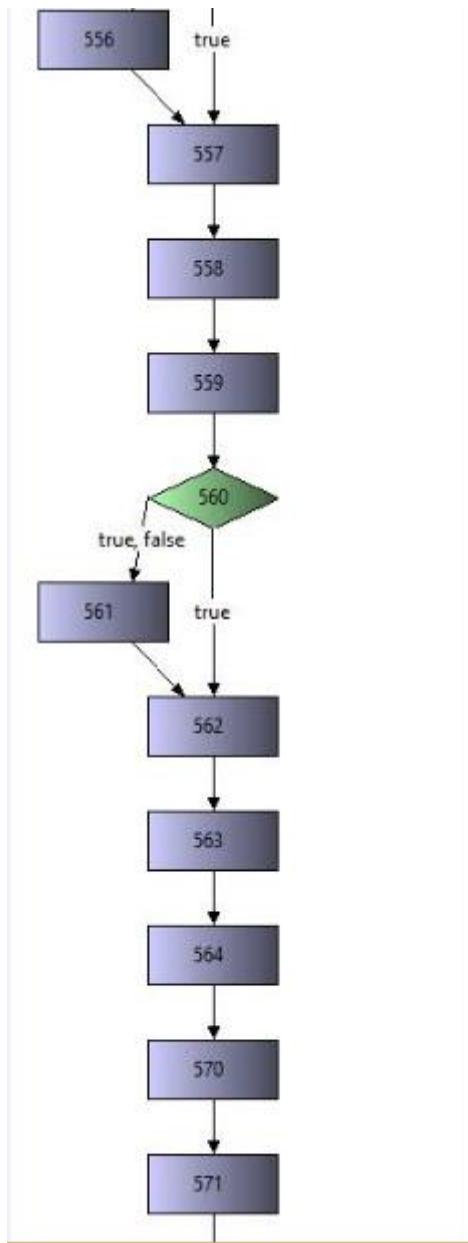
531 public static int submitReport() {
532
533     int pain, drowsiness, nausea, anxiety, depression;
534
535     System.out.println("Enter the severity of your symptoms from 1-10.");
536     Scanner sc = new Scanner(System.in);
537     System.out.println("Enter severity of Pain: ");
538     pain = sc.nextInt();
539     if (pain >10 || pain <1){
540         System.out.println("Enter Proper Value");
541         pain = sc.nextInt();
542     }
543     sc.nextLine();
544     System.out.println("Enter severity of Drowsiness: ");
545     drowsiness = sc.nextInt();
546     if (drowsiness >10 || drowsiness <1)
547         drowsiness = sc.nextInt();
548     sc.nextLine();
549     System.out.println("Enter severity of Nausea: ");
550     nausea = sc.nextInt();
551     if (nausea >10 || nausea <1)
552         nausea = sc.nextInt();
553     sc.nextLine();
554     System.out.println("Enter severity of Anxiety: ");
555     anxiety = sc.nextInt();
556     if (anxiety >10 || anxiety <1)
557         anxiety = sc.nextInt();
558     sc.nextLine();
559     System.out.println("Enter severity of Depression: ");
560
561     depression = sc.nextInt();
562     if (depression >10 || depression <1)
563         depression = sc.nextInt();
564     sc.nextLine();
565     System.out.println("Enter any further comments.");
566     String comments = sc.nextLine();
567
568     try
569     {
570         // create mysql database connection
571         String myDriver = "org.gjt.mm.mysql.Driver";
572         String myUrl = URL;
573         String user = USERNAME;
574         String password = PASSWORD;
575         Class.forName(myDriver);
576         Connection conn = DriverManager.getConnection(myUrl, user, password);
577
578         //DateFormat df = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss");
579         Date date = new Date();
580
581         //String date = df.format(d);
582         SimpleDateFormat dt = new SimpleDateFormat("yyyyMMddhhmmss");
583
584         // SQL INSERT query.
585
586         String query = "INSERT INTO reports (patient, dateTime, pain, drowsiness, nausea, anxiety, depression, commen
587

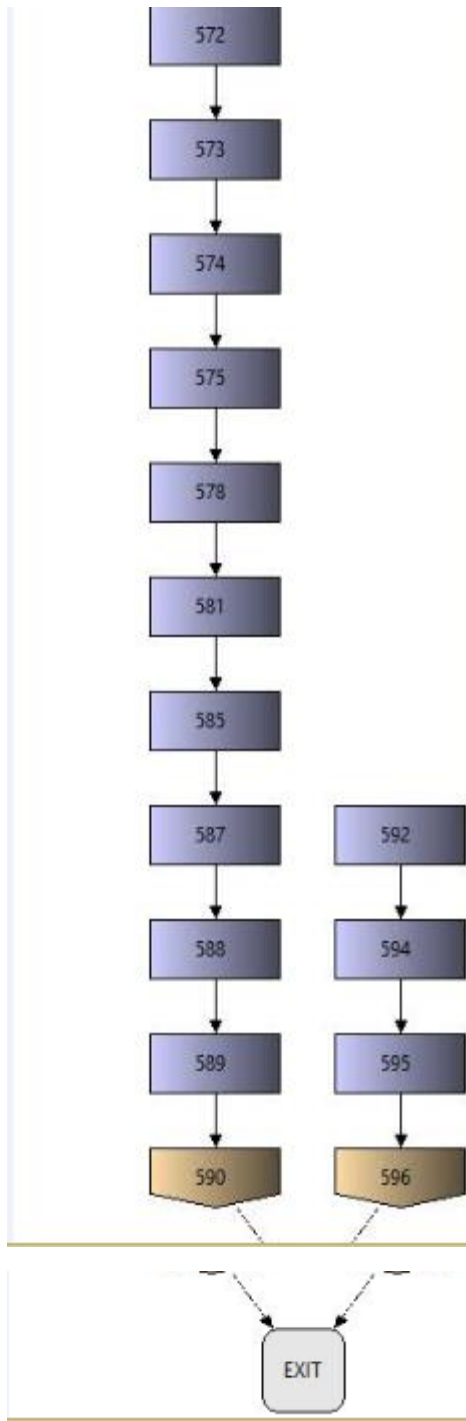
```

```
589 Statement st = conn.createStatement();
590 st.executeUpdate(query);
591 st.close();
592 return 1;
593 }
594 catch (Exception e)
595 {
596     System.err.println("Got an exception! ");
597     System.err.println(e.getMessage());
598     return 0;
599 }
600 // INSERT INTO reports (pain, drowsiness, nausea, anxiety, depression) VALUES ("pain+", "+drowsiness+", "+nausea
601
602 }
603
604 }
```









Paths: 5+1=6

Path 1:534-535-536-537-538-542-543-544-545-true-547-548-549-550-true-552-553-554-555-557-558-559-560-562-564-570-571-572-573-574-575-578-581-585-587-588-589-590

Path 2:534-535-536-537-538-539-540-542-543-544-545-true-547-548-549-550-true-552-553-554-555-557-558-559-560-562-564-570-571-572-573-574-575-578-581-585-587-588-589-590

Path 3: 534-535-536-537-538-539-540-542-543-544-545-false-546-547-548-549-550-true-552-553-554-555-557-558-559-560-562-564-570-571-572-573-574-575-578-581-585-587-588-589-590