Algorithm (User side)

1. Start
2. Enter 1st, 2nd and 3rd symptom
3. Read matched department output
4. End

Algorithm (System side)

1. Start
2. Read 1st,2nd and 3rd symptom
3. If 1st and 2nd symptoms match
   1. Then
      1. Read DB and summon the general case those two symptoms are indicative of, placeholder name DB\_gc\_A
      2. If DB\_gc\_A and 3rd symptom match
         1. Then
            1. Read DB and summon the potential disease that the general case and third symptom are indicative of, Placeholder name DB\_pd\_X
            2. Read DB and summon department for DB\_pd\_X
            3. Print department
         2. Else
            1. Read DB and summon the general case DB\_gc\_A
            2. Read DB and summon department for DB\_gc\_A
            3. Print department
   2. Else if 2nd and 3rd symptoms match
      1. Then
         1. Read DB and summon the general case those two symptoms are indicative of, placeholder name DB\_gc\_B
         2. If DB\_gc\_B and 1st symptom match
            1. Then

Read DB and summon the potential disease that the general case and third symptom are indicative of, Placeholder name DB\_pd\_Y

Read DB and summon department for DB\_pd\_Y

Print department

* + - * 1. Else

Read DB and summon the general case DB\_gc\_B

Read DB and summon department for DB\_gc\_B

Print department

* + 1. Else if 1st and 3rd symptoms match
       1. Then
          1. Read DB and summon the general case those two symptoms are indicative of, placeholder name DB\_gc\_C
          2. If DB\_gc\_C and 2nd symptom match

Then

Read DB and summon the potential disease that the general case and symptom are indicative of, Placeholder name DB\_pd\_Z

Read DB and summon department for DB\_pd\_Z

Print department

Else

Read DB and summon the general case DB\_gc\_C

Read DB and summon department for DB\_gc\_C

Print department

* + - 1. Else
         1. Print “Could not find a suitable department, please check with a general physician”