CPT111 PRINCIPLES OF PROGRAMMING

Pusat Pengajian Sains Komputer

Universiti Sains Malaysia

Semester I 2020/2021

Course: CPT111

Assignment: Hackathon 3

Session: Sem 1 2021/2022

Date: 31 December 2021

Group Number: 44

Member List:

1. YEO YING SHENG (157627)
2. CHALLVEN JAPIRIN (157698)
3. EDU SINUSI (159136)

Lecturer's Name: Pn Maziani Sabudin

Table of Content

1.0) Problem Analysis

Develop a C++ program to help a small private hospital the hospital billing system for their patients. The program will compute a patient's bill for the hospital using user-defines functions. The program will calculate the total of patient's charges which includes hospital stay, surgery charges, pharmacy charges and other services charges. It will also keep track of the number of days the patient stays in the hospital and find the total rates based on the number of days. The system also calculates the hospital's daily rate based on room types and food charges. The program will show surgery charges for at least five (5) types of surgery, pharmacy charges for at least five (5) types of medication and types of service such as X-ray/radiology, laboratory test, physiotherapy that the patient might take. All the types of surgery, pharmacy and services have different charges based on the patient's choice. The total charges will be displayed after the user inputs all the patient's bills based on their hospital stay, surgery charges, pharmacy charges and other services charges.

2.0) Specification Requirement

2.1) Input:

1. The choice of number option based on the Patient's Billing Menu.
2. The status of a patient, either an inpatient or an outpatient.
3. Types of room of a patient.
4. Number of days that a patient has checked into the hospital.
5. Sets of meals received by a patient.
6. Types of surgery received by a patient.
7. Types of medication received by a patient.
8. Number of doses of medicine per day taken by a patient.
9. Number of days the medicine is taken by a patient.
10. Types of services received by a patient.
11. Number of times a patient receives services.

2.2) Process:

1. Repeat reading and adding the total of patient's bill based on the Patient Billing Menu.
2. Determine the price of the room based on the type of room of the patient.
3. Calculate the bill based on how many days the patient has checked into the hospital.
4. Calculate the total cost of the sets of meals based on different sets of meals received by a patient each day.
5. Calculate the hospital stay charges by multiplying the price of the room and stay duration, and then adding the result with food fee.

dailyStay = roomTypes * stayDuration + (setA + setB + setC) (optional)

6. Determine the price of each surgery received by the patient.
7. Calculate the surgery charges based on different surgeries of the patient.
8. Determine the price of each dose of medicine received by the patient.
9. Calculate the pharmacy charges based on the number of doses and the number of days that the medicine has been taken by the patient.
10. Determine the price of each service received by the patient.

11. Calculate the services charges based on the number of times that different services are received by the patient.
12. Calculate the final payment of the patient's bill by adding together the hospital stays charges, surgery charges, pharmacy charges, and services charges.

totalCost = hospitalStayCost + surgeryCost + pharmacyCost + serviceCost

2.3) Output:

1. Final payment of the patient's bill.

2.4) Constraints and Assumptions

1. A patient must be an in-patient to pay for the hospital stays charges which include the types of rooms, duration of hospital stays and different sets of meals.
2. Hospital stays charges must be calculated by multiplying the price of room with the number of days spent by a patient in the hospital and then added together with the price of each set of meal received by the patient.
3. The cost of physiotherapy must be multiplied with the number of hours of physiotherapy received by a patient to calculate the real charges of physiotherapy sessions.
4. An administrator or a user must enter an appropriate number which is within the range of the numbers displayed on the menu.
5. The cost of each medicine must be calculated by multiplying the original price of each pharmaceutical medicine with the number of doses and the number of days taken by a patient.
6. The services charges must be calculated according to the number of services received by a patient in the hospital.
7. The global variables are all constant.

3.0 Variable Definitions

Variables	Definitions
FIRST_CHOICE	A constant global variable which holds the integer 1
SECOND_CHOICE	A constant global variable which holds the integer 2
THIRD_CHOICE	A constant global variable which holds the integer 3
FOURTH_CHOICE	A constant global variable which holds the integer 4
FIFTH_CHOICE	A constant global variable which holds the integer 5
SIXTH_CHOICE	A constant global variable which holds the integer 6
EXIT_DECISION	A constant global variable which holds the integer 7
CHECKOUT	A constant local variable which holds the integer 5
QUIT_FOOD_MENU	A constant local variable which holds the integer 5
NO_FOOD_ORDERED	A constant local variable which holds the integer 4
patientStat	To represent an in-patient or an out-patient
foodTypes	To represent the types of food of a patient
setA	To represent the cost of a set of meal in the hospital
setB	To represent the cost of a set of meal in the hospital
setC	To represent the cost of a set of meal in the hospital
deluxeSuiteFee	To represent the cost of a deluxe suite per day
singleDeluxeFee	To represent the cost of a single deluxe room per day
singleStandardFee	To represent the cost of a single standard room per day
twoBeddedFee	To represent the cost of a two bedded room per day
fourBeddedFee	To represent the cost of a four bedded room per day
icuFee	To represent the cost of an ICU per day
roomTypes	To represent different types of room of a patient
stayDuration	To represent the duration of hospital stays
surgeryFees	Sum of surgery charges to be paid by the patient
spineFee	Surgery charge for spine surgery

Variables	Definitions
hipReplacementFee	Surgery charge for hip replacement surgery
coronaryBypassFee	Surgery charge for hip replacement surgery
angioplastyFee	Surgery charge for angioplasty surgery
kneeReplacementFee	Surgery charge for knee replacement surgery
kidneyStoneRemovalFee	Surgery charge for kidney stone removal surgery
option	Holds the choice made by the program user
cntSurgery	Acts as a counter
medicationFees	Sum of pharmacy charges to be paid by the patient
paracetamolFee	Pharmacy charge of one dose of paracetamol
chlorpheniramineFee	Pharmacy charge of one dose of chlorpheniramine
cetirizineFee	Pharmacy charge of one dose of cetirizine
diphenhydramineFee	Pharmacy charge of one dose of diphenhydramine
loratadineFee	Pharmacy charge of one dose of loratadine
diclofenacFee	Pharmacy charge of one dose of diclofenac
doses	Number of doses of medicine taken by the patient
medDay	Number of days a patient takes his or her medicine
serviceFreq	Number of times a patient receives hospital services
serviceTypes	To represent different types of service received by the patient
physioDuration	The duration that a patient undergoes a physiotherapy
serviceFees	Sum of service charges to be paid by the patient
xRayFee	Service charge of X-ray
bloodTestFee	Service charge of blood test
mriFee	Service charge of MRI scan

Variables	Definitions
ctFee	Service charge of CT scan
physiotherapyFee	Service charge of physiotherapy session per hour
vaccinationFee	Service charge of vaccination
decision	Choice made by program user in the main menu
hospitalStayCost	Hospital stays charges
surgeryCost	Surgery charges
pharmacyCost	Pharmacy charges
serviceCost	Services charges
totalCost	The sum of hospital stays charges, surgery charges, pharmacy charges and services charges
dailyRate	To represent the hospital stays charges according to the types of rooms, duration of stays and different sets of meals

4.0) Pseudocode

START

1.0 INIT global constant variable FIRST_CHOICE = 1, SECOND_CHOICE = 2, THIRD_CHOICE = 3, FOURTH_CHOICE = 4, FIFTH_CHOICE = 5, SIXTH_CHOICE = 6, EXIT_ DECISION = 7

2.0 FUNC void showMenu ( )

2.1 INIT constant variable CHECKOUT = 5

2.2 Display patient billing menu

ENDFUNC

3.0 FUNC void patientStatus ( )

3.1 Display patient status

ENDFUNC

4.0 FUNC void patientDetails (&roomTypes, &dailyRate, &stayDuration)

4.1 INIT constant variable QUIT_FOOD_MENU = 5, NO_FOOD_ORDERED = 4

INIT variable setA = 5.00, setB = 4.00, setC = 4.50, deluxeSuiteFee = 700.00, singleDeluxeFee = 338.00, singleStandardFee = 268.00, twoBeddedFee = 150.00, fourBeddedFee = 95.00, icuFee = 400.00

4.2 CALL patientStatus ( )

4.3 READ patientStat

system (“cls”)

4.4 WHILE (patientStat < FIRST_CHOICE | | patientStat > SECOND_CHOICE)

4.4.1 CALL patientStatus( )

4.4.2 READ patientStat

system (“cls”)

ENDWHILE

4.5 IF (patientStat ! = 1)

4.5.1 print message “The patient is not registered as an in-patient, thus no information regarding the patient.”

RETURN

ENDIF

4.6 READ roomTypes

system (“cls”)

4.7 WHILE (roomTypes < FIRST_CHOICE | | roomTypes > EXIT_DECISION)

4.7.1 READ roomTypes

ENDWHILE

4.8 READ stayDuration

system (“cls”)

4.9 CASE OF roomTypes

4.9.1 FIRST_CHOICE : dailyRate += deluxeSuiteFee

4.9.2 SECOND_CHOICE : dailyRate += singleDeluxeFee

4.9.3 THIRD_CHOICE : dailyRate += singleStandardFee

4.9.4 FOURTH_CHOICE : dailyRate += twoBeddedFee

4.9.5 FIFTH_CHOICE : dailyRate += fourBeddedFee

4.9.6 SIXTH_CHOICE : dailyRate += icuFee

4.9.7 OTHERS : default

ENDCASE

4.10 dailyRate *= stayDuration

4.11 FOR (int i = 1; i < = stayDuration; i ++)

4.11.1 READ foodTypes

system (“cls”)

4.11.1.1 WHILE (foodTypes < FIRST_CHOICE | | foodTypes > QUIT_FOOD_MENU)

4.11.1.1.1 READ foodTypes

system (“cls”)

ENDWHILE

4.11.3 IF (foodTypes == NO_FOOD_ORDERED)

CONTINUE

4.11.4 ELSE IF (foodTypes == QUIT_FOOD_MENU)

RETURN

4.11.5 ELSE

4.11.5.1 CASE OF foodTypes

4.11.5.1.1 FIRST_CHOICE : dailyRate += setA

4.11.5.1.2 SECOND_CHOICE : dailyRate += setB

4.11.5.1.3 THIRD_CHOICE : dailyRate += setC

4.11.5.1.4 OTHERS : default

ENDCASE

ENDIF

ENDFOR

ENDFUNC

5.0 FUNC double hospitalStay ( )

5.1 INIT roomTypes = 0, stayDuration = 0, dailyRate = 0

5.2 CALL patientDetails (roomTypes, dailyRate, stayDuration)

5.3 RETURN dailyRate

END FUNC

6.0 FUNC double surgeryCharges ( )

6.1 INIT surgeryFees = 0, spineFee = 55000.00, hipReplacementFee = 50000.00, coronaryBypassFee = 30000.00, angioplastyFee = 20000.00, kneeReplacementFee = 35000.00, kidneyStoneRemovalFee = 15000.00, cntSurgery = 0

6.2 DO WHILE (option ! = EXIT_DECISION)

6.2.1 IF (cntSurgery == 0)

6.2.1.1 READ option

system (“cls”)

6.2.1.2 cntSurgery ++

6.2.2 ELSE

6.2.2.1 READ option

system (“cls”)

ENDIF

6.2.3 WHILE (option < FIRST_CHOICE | | option > EXIT_DECISION)

6.2.3.1 READ option

system (“cls”)

ENDWHILE

6.2.4 CASE OF option

6.2.4.1 FIRST_CHOICE : surgeryFees += spineFee

6.2.4.2 SECOND_CHOICE : surgeryFees += hipReplacementFee

6.2.4.3 THIRD_CHOICE : surgeryFees += coronaryBypassFee

6.2.4.4 FOURTH_CHOICE : surgeryFees += angioplastyFee

6.2.4.5 FIFTH_CHOICE : surgeryFees += kneeReplacementFee

6.2.4.6 SIXTH_CHOICE : surgeryFees += kidneyStoneRemovalFee

6.2.4.7 OTHERS: default

ENDCASE

END DO-WHILE LOOP

6.3 RETURN surgeryFees

ENDFUNC

7.0 FUNC double pharmacyCharges ( )

7.1 INIT medicationFees = 0, paracetamolFee = 13.00, chlorpheniramineFee = 20.00, cetirizineFee = 15.00, diphenhydramineFee = 10.00, loratadineFee = 11.00, diclofenacFee = 19.00, cntPharm = 0

7.2 DO WHILE (option ! = EXIT_DECISION)

7.2.1 IF (cntPharm == 0)

7.2.1.1 READ option

7.2.1.2 cntPharm++

7.2.2 ELSE

7.2.2.1 READ option

ENDIF

7.2.3 WHILE (option < FIRST_CHOICE | | option > EXIT_DECISION)

system (“cls”)

7.2.3.1 READ option

ENDWHILE

7.2.4 IF (option ! = EXIT_DECISION)

7.2.4.1 READ doses

7.2.4.2 READ medDay

system (“cls”)

7.2.5 ELSE

system (“cls”)

ENDIF

7.2.6 CASE OF option

7.2.6.1 FIRST CHOICE : medicationFees += (paracetamolFee * doses * medDay)

7.2.6.2 SECOND_CHOICE : medicationFees += (chlorpheniramineFee * doses * medDay)

7.2.6.3 THIRD_CHOICE : medicationFees += (cetirizineFee * doses * medDay)

7.2.6.4 FOURTH_CHOICE : medicationFees += (diphenhydramineFee * doses * medDay)

7.2.6.5 FIFTH_CHOICE : medicationFees += (loratadineFee * doses * medDay)

7.2.6.6 SIXTH_CHOICE : medicationFee += (diclofenacFee * doses * medDay)

7.2.6.7 OTHERS : default

ENDCASE

END DO-WHILE LOOP

7.3 RETURN medicationFees

ENDFUNC

8.0 FUNC double serviceCharges ( )

8.1 INIT serviceFees = 0, xRayFee = 80.00, bloodTestFee = 100.00, mriFee = 1200.00, ctFee = 1000.00, physiotheraphyFee = 170.00, vaccinationFee = 70.00

8.2 READ serviceFreq

8.3 FOR (int i = 1; i < = serviceFreq; i ++)

8.3.1 READ serviceTypes

system (“cls”)

8.3.2 WHILE (serviceTypes < FIRST_CHOICE | | serviceTypes > EXIT_DECISION)

8.3.2.1 READ serviceTypes

system (“cls”)

ENDWHILE

8.3.3 CASE OF serviceTypes

8.3.3.1 FIRST_CHOICE : serviceFees += xRayFee

8.3.3.2 SECOND_CHOICE : serviceFees += bloodTestFee

8.3.3.3 THIRD_CHOICE : serviceFees += mriFee

8.3.3.4 FOURTH_CHOICE : serviceFees += ctFee

8.3.3.5 FIFTH_CHOICE : READ physioDuration

serviceFees += (physiotherapyFee * physioDuration)

8.3.3.6 SIXTH_CHOICE : serviceFees += vaccinationFee

8.3.3.7 OTHERS : default

ENDCASE

ENDFOR

8.4 RETURN serviceFees

ENDFUNC

9.0 FUNC double patientBill (hospitalStayCost, surgeryCost, pharmacyCost, serviceCost)

9.1 RETURN hospitalStayCost + surgeryCost + pharmacyCost + serviceCost

ENDFUNC

10.0 START Main function

10.1 INIT constant variable CHECKOUT = 5

10.2 DO WHILE (decision ! = CHECKOUT)

10.2.1 CALL showMenu ( )

10.2.2 READ decision

system (“cls”)

10.2.3 WHILE (decision > CHECKOUT | | decision < FIRST_CHOICE)

10.2.3.1 CALL showMenu ( )

10.2.3.2 READ decision

system (“cls”)

ENDWHILE

10.2.4 IF (decision ! = CHECKOUT)

10.2.4.1 CASE FOR decision

10.2.4.1.1 FIRST_CHOICE : hospitalStayCost = CALL hospitalStay ( )

10.2.4.1.2 SECOND_CHOICE : surgeryCost = CALL surgeryCharges ( )

10.2.4.1.3 THIRD CHOICE : pharmacyCost = CALL pharmacyCharges ( )

10.2.4.1.4 FOURTH_CHOICE : serviceCost = CALL

serviceCharges ( )

10.2.4.1.5 OTHERS : default

ENDCASE

ENDIF

END DO-WHILE LOOP

10.3 totalCost = CALL patientBill (hospitalStayCost, surgeryCost, pharmacyCost, serviceCost)

10.4 PRINT totalCost

10.5 PRINT message “The patient has been checked out”

STOP

5.0) Source Code

```
/*
*This program has been developed to help a small private hospital and oversee the hospital billing system for their patients. It
computes a patient's bill for the hospital using user-defined functions.
*Developers: Yeo Ying Sheng, Edu Sinusi, Challven Japirin
*Contact: yeousm@student.usm.my, sinusiedu@student.usm.my, challven001@student.usm.my
*/

#include <iostream>
#include <iomanip>
#include <cmath>
#include <string>

using namespace std;

//Declare and define constant global variables
const int   FIRST_CHOICE = 1,
            SECOND_CHOICE = 2,
            THIRD_CHOICE = 3,
            FOURTH_CHOICE = 4,
            FIFTH_CHOICE = 5,
            SIXTH_CHOICE = 6,
            EXIT_DECISION = 7;


//Display the main menu
void showMenu()
{
    const int CHECKOUT = 5; //Declare and define constant local variable
    cout << "Patient Billing Menu\n"
         << "--------------------------------\n"
         << "1. Hospital Daily Rate\n"
         << "2. Types of Surgery\n"
         << "3. Types of Medication\n"
         << "4. Types of Service\n"
         << "5. Check out\n"
         << "--------------------------------\n";
}


//Display the status of a patient
void patientStatus()
{
    cout << "Patient Status\n"
         << "--------------------\n"
         << "1. In-patient\n"
         << "2. Out-patient\n"
         << "--------------------\n"
         << "Status of patient: ";
}


//Calculate the payment based on types of room, duration that the patient has been staying in the hospital and different sets of meals
void patientDetails(int &roomTypes, double &dailyRate, int &stayDuration)

{
    const int   QUIT_FOOD_MENU = 5,
                NO_FOOD_ORDERED = 4;
    int foodTypes, patientStat;
    double  setA = 5.00, //Cost of each set of meal in the hospital
            setB = 4.00,
            setC = 4.50,
            deluxeSuiteFee = 700.00, //Cost of each room per day
            singleDeluxeFee = 338.00,
            singleStandardFee = 268.00,
            twoBeddedFee = 150.00,
            fourBeddedFee = 95.00,
            icuFee = 400.00;

    patientStatus();
    cin >> patientStat;
    system("cls");
```

```
71:
72:     //To determine if the patientStat is within the range
73:     while(patientStat < FIRST_CHOICE || patientStat > SECOND_CHOICE)
74:     {
75:         patientStatus();
76:         cin >> patientStat;
77:         system("cls");
78:     }
79:
80:     //To determine whether the patient is an in-patient or an out-patient
81:     if(patientStat != 1)
82:     {
83:         cout << "The patient is not registered as an in-patient, thus no information
    regarding the patient.\n\n";
84:         return;
85:     }
86:
87:     cout << "Room Categories\n"
88:          << "--------------------------------\n"
89:          << "1. Deluxe Suite\n"
90:          << "2. Single Deluxe\n"
91:          << "3. Single Standard\n"
92:          << "4. Two Bedded\n"
93:          << "5. Four Bedded\n"
94:          << "6. ICU\n"
95:          << "7. Return to Menu\n"
96:          << "--------------------------------\n"
97:          << "Please pick an option: ";
98:     cin >> roomTypes;
99:     system("cls");
100:
101:    //To determine if the roomTypes is within the range
102:    while(roomTypes < FIRST_CHOICE || roomTypes > EXIT_DECISION)
103:    {
104:        cout << "Please pick a valid option: ";
105:        cin >> roomTypes;
106:    }
107:


108:    cout << "How many days has the patient checked into the hospital: ";
109:    cin >> stayDuration;
110:    system("cls");
111:
112:    //To calculate the hospital stays charges based on the price of room
113:    switch (roomTypes)
114:    {
115:        case FIRST_CHOICE:
116:        {
117:            dailyRate += deluxeSuiteFee;
118:            break;
119:        }
120:        case SECOND_CHOICE:
121:        {
122:            dailyRate += singleDeluxeFee;
123:            break;
124:        }
125:        case THIRD_CHOICE:
126:        {
127:            dailyRate += singleStandardFee;
128:            break;
129:        }
130:        case FOURTH_CHOICE:
131:        {
132:            dailyRate += twoBeddedFee;
133:            break;
134:        }
135:        case FIFTH_CHOICE:
136:        {
137:            dailyRate += fourBeddedFee;
138:            break;
139:        }
```

```
140:         case SIXTH_CHOICE:
141:         {
142:             dailyRate += icuFee;
143:             break;
144:         }
145:         default:
146:             break;
147:     }
148:
149:     //Multiply the dailyRate that contains the room price with the number of days
     that a patient has been staying in the hospital
150:     dailyRate *= stayDuration;
151:
152:     //Calculate the hospital stays charges by adding the price of each meal set with
     dailyRate
153:     for(int i=1; i <= stayDuration; i++)
154:     {
155:         cout << "Food Menu\n"
156:              << "--------------------------------\n"
157:              << "1. Set A\n"
158:              << "2. Set B\n"
159:              << "3. Set C\n"
160:              << "4. No food ordered\n"

161:              << "5. Return to Menu\n"
162:              << "--------------------------------\n"
163:              << "Please pick the set of meal on day " << i <<  ": ";
164:         cin >> foodTypes;
165:         system("cls");
166:
167:         while(foodTypes < FIRST_CHOICE || foodTypes > QUIT_FOOD_MENU)
168:         {
169:             cout << "Food Menu\n"
170:                  << "--------------------------------\n"
171:                  << "1. Set A\n"
172:                  << "2. Set B\n"
173:                  << "3. Set C\n"
174:                  << "4. No food ordered\n"
175:                  << "5. Return to Menu\n"
176:                  << "--------------------------------\n"
177:                  << "Please pick a valid option: ";
178:             cin >> foodTypes;
179:             system("cls");
180:         }
181:
182:         if(foodTypes == NO_FOOD_ORDERED)
183:         {
184:             continue;
185:         }
186:         else if(foodTypes == QUIT_FOOD_MENU)
187:         {
188:             return;
189:         }
190:         else
191:         {
192:             switch(foodTypes)
193:             {
194:                 case FIRST_CHOICE:
195:                 {
196:                     dailyRate += setA;
197:                     break;
198:                 }
199:                 case SECOND_CHOICE:
200:                 {
201:                     dailyRate += setB;
202:                     break;
203:                 }
204:                 case THIRD_CHOICE:
205:                 {
206:                     dailyRate += setC;
207:                     break;
208:                 }
```

```
                default:
                    break;
            }
        }
    }
}


//Call the function and return the hospital stays charges of a patient
double hospitalStay()
{
    int roomTypes = 0,
        stayDuration = 0;

    double  dailyRate = 0;

    patientDetails(roomTypes, dailyRate, stayDuration);

    return dailyRate;
}


//Calculate and return the amount of surgery charges of a patient
double surgeryCharges()
{
    double  surgeryFees = 0,
            spineFee = 55000.00,
            hipReplacementFee = 50000.00,
            coronaryBypassFee = 30000.00,
            angioplastyFee = 20000.00,
            kneeReplacementFee = 35000.00,
            kidneyStoneRemovalFee = 15000.00;

    int option, cntSurgery = 0;

    do
    {
        if(cntSurgery == 0)
        {
            cout << "Surgery Categories\n"
                 << "--------------------------------\n"
                 << "1. Spine Surgery\n"
                 << "2. Hip Replacement Surgery\n"
                 << "3. Coronary Bypass Surgery\n"
                 << "4. Angioplasty Surgery\n"
                 << "5. Knee Replacement Surgery\n"
                 << "6. Kidney Stone Removal Surgery\n"
                 << "7. Return to Menu\n"
                 << "--------------------------------\n"
                 << "Please pick an option: ";
            cin >> option;
            system("cls");
            cntSurgery++;
        }
        else
        {
            cout << "Surgery Categories\n"
                 << "--------------------------------\n"
                 << "1. Spine Surgery\n"
                 << "2. Hip Replacement Surgery\n"
                 << "3. Coronary Bypass Surgery\n"
                 << "4. Angioplasty Surgery\n"

                 << "5. Knee Replacement Surgery\n"
                 << "6. Kidney Stone Removal Surgery\n"
                 << "7. Return to Menu\n"
                 << "--------------------------------\n"
                 << "Please pick another option (Press 7 to return to menu): ";
            cin >> option;
            system("cls");
        }

```

```
280:         while(option < FIRST_CHOICE || option > EXIT_DECISION)
281:         {
282:             cout << "Surgery Categories\n"
283:                  << "--------------------------------\n"
284:                  << "1. Spine Surgery\n"
285:                  << "2. Hip Replacement Surgery\n"
286:                  << "3. Coronary Bypass Surgery\n"
287:                  << "4. Angioplasty Surgery\n"
288:                  << "5. Knee Replacement Surgery\n"
289:                  << "6. Kidney Stone Removal Surgery\n"
290:                  << "7. Return to Menu\n"
291:                  << "--------------------------------\n"
292:                  << "Please pick a valid option: ";
293:             cin >> option;
294:             system("cls");
295:         }
296:
297:         switch (option)
298:         {
299:             case FIRST_CHOICE:
300:             {
301:                 surgeryFees += spineFee;
302:                 break;
303:             }
304:             case SECOND_CHOICE:
305:             {
306:                 surgeryFees += hipReplacementFee;
307:                 break;
308:             }
309:             case THIRD_CHOICE:
310:             {
311:                 surgeryFees += coronaryBypassFee;
312:                 break;
313:             }
314:             case FOURTH_CHOICE:
315:             {
316:                 surgeryFees += angioplastyFee;
317:                 break;
318:             }
319:             case FIFTH_CHOICE:
320:             {
321:                 surgeryFees += kneeReplacementFee;
322:                 break;
323:             }
324:             case SIXTH_CHOICE:
325:             {

326:                 surgeryFees += kidneyStoneRemovalFee;
327:                 break;
328:             }
329:             default:
330:                 break;
331:         }
332:     }while(option != EXIT_DECISION);
333:
334:     return surgeryFees;
335: }
336:
337:
338: //Calculate and return the pharmacy charges of a patient based on the number of
     doses of medicine and the number of days taken by a patient
339: double pharmacyCharges()
340: {
341:     double  medicationFees = 0,
342:             paracetamolFee = 13.00, //cost per dose
343:             chlorpheniramineFee = 20.00,
344:             cetirizineFee = 15.00,
345:             diphenhydramineFee = 10.00,
346:             loratadineFee = 11.00,
347:             diclofenacFee = 19.00;
348:
349:     int option, doses, medDay, cntPharm = 0;
```

```
350:
351:     do
352:     {
353:         if(cntPharm == 0)
354:         {
355:             cout << "Medication Categories\n"
356:                  << "--------------------------------\n"
357:                  << "1. Paracetamol\n"
358:                  << "2. Chlorpheniramine\n"
359:                  << "3. Cetirizine\n"
360:                  << "4. Diphenhydramine\n"
361:                  << "5. Loratadine\n"
362:                  << "6. Diclofenac\n"
363:                  << "7. Return to Menu\n"
364:                  << "--------------------------------\n"
365:                  << "Please pick an option: ";
366:             cin >> option;
367:             cntPharm++;
368:         }
369:         else
370:         {
371:             cout << "Medication Categories\n"
372:                  << "--------------------------------\n"
373:                  << "1. Paracetamol\n"
374:                  << "2. Chlorpheniramine\n"
375:                  << "3. Cetirizine\n"
376:                  << "4. Diphenhydramine\n"
377:                  << "5. Loratadine\n"
378:                  << "6. Diclofenac\n"
379:                  << "7. Return to Menu\n"

380:                  << "--------------------------------\n"
381:                  << "Please pick another option (Press 7 to return to menu): ";
382:             cin >> option;
383:         }
384:
385:
386:
387:         while(option < FIRST_CHOICE || option > EXIT_DECISION)
388:         {
389:             system("cls");
390:             cout << "Medication Categories\n"
391:                  << "--------------------------------\n"
392:                  << "1. Paracetamol\n"
393:                  << "2. Chlorpheniramine\n"
394:                  << "3. Cetirizine\n"
395:                  << "4. Diphenhydramine\n"
396:                  << "5. Loratadine\n"
397:                  << "6. Diclofenac\n"
398:                  << "7. Return to Menu\n"
399:                  << "--------------------------------\n"
400:                  << "Please pick a valid option: ";
401:             cin >> option;
402:         }
403:
404:         if(option != EXIT_DECISION)
405:         {
406:             cout << "Number of doses of medicine per day taken by the patient: ";
407:             cin >> doses;
408:             cout << "Number of days the medicine is taken by the patient: ";
409:             cin >> medDay;
410:             cout << endl;
411:             system("cls");
412:         }
413:         else
414:         {
415:             system("cls");
416:         }
417:
```

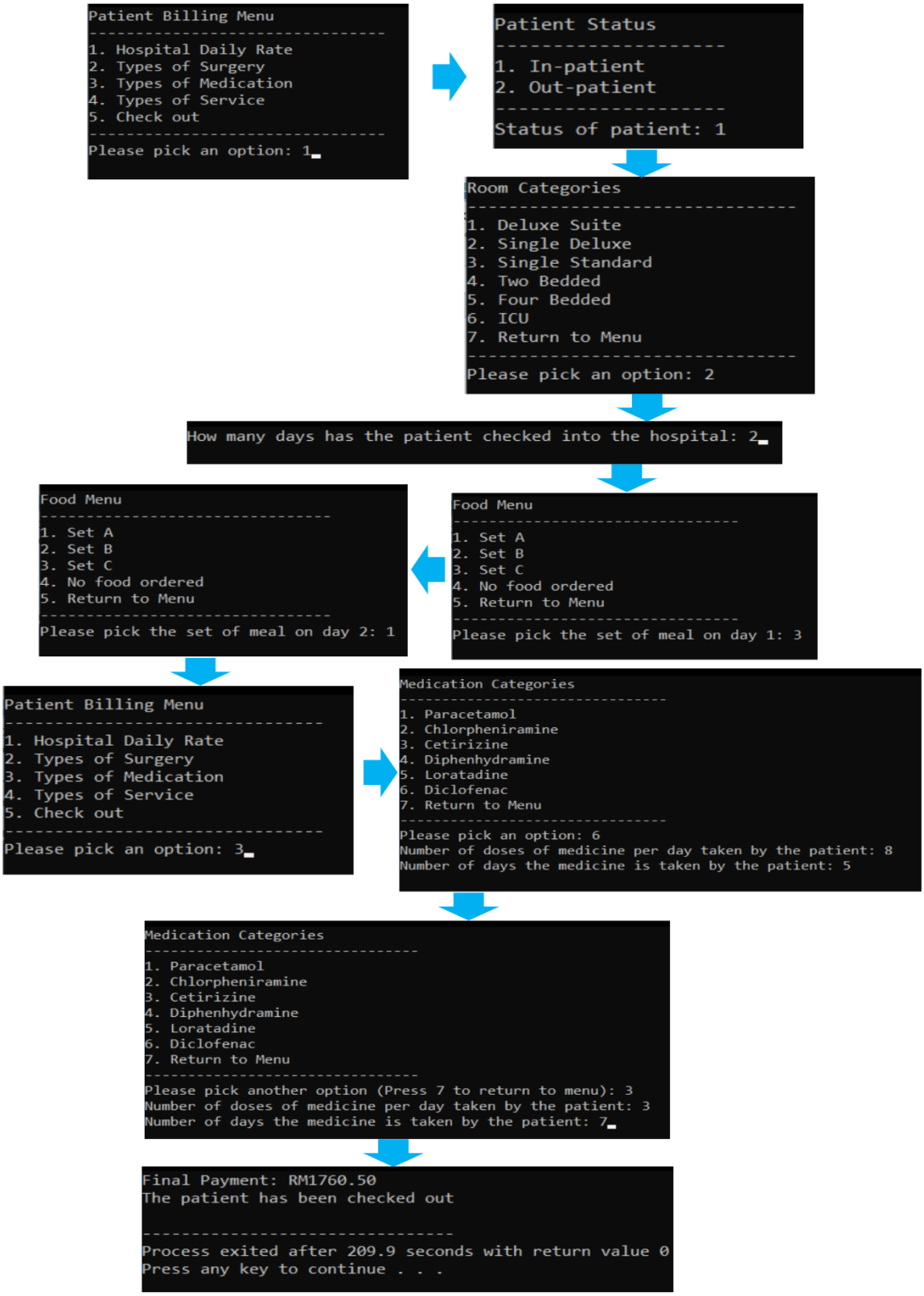
```
418:         switch (option)
419:         {
420:             case FIRST_CHOICE:
421:             {
422:                 medicationFees += (paracetamolFee*doses*medDay);
423:                 break;
424:             }
425:             case SECOND_CHOICE:
426:             {
427:                 medicationFees += (chlorpheniramineFee*doses*medDay);
428:                 break;
429:             }
430:             case THIRD_CHOICE:
431:             {
432:                 medicationFees += (cetirizineFee*doses*medDay);
433:                 break;
434:             }

435:             case FOURTH_CHOICE:
436:             {
437:                 medicationFees += (diphenhydramineFee*doses*medDay);
438:                 break;
439:             }
440:             case FIFTH_CHOICE:
441:             {
442:                 medicationFees += (loratadineFee*doses*medDay);
443:                 break;
444:             }
445:             case SIXTH_CHOICE:
446:             {
447:                 medicationFees += (diclofenacFee*doses*medDay);
448:                 break;
449:             }
450:             default:
451:                 break;
452:         }
453:     }while(option != EXIT_DECISION);
454: 
455:     return medicationFees;
456: }
457: 
458: 
459: //Calculate and return the service charges of a patient for using the services of
    hospital
460: double serviceCharges()
461: {
462:     int serviceFreq, serviceTypes;
463:     double  physioDuration,
464:             serviceFees = 0,
465:             xRayFee = 80.00,
466:             bloodTestFee = 100.00,
467:             mriFee = 1200.00,
468:             ctFee = 1000.00,
469:             physiotherapyFee = 170.00, //Cost of physiotherapy per hour
470:             vaccinationFee = 70.00;
471: 
472:     cout << "How many times has the patient received services: ";
473:     cin >> serviceFreq;
474:     cout << endl;
475: 
476:     for(int i=1; i <= serviceFreq; i++)
477:     {
478:         cout << "Service Categories\n"
479:              << "------------------------------\n"
480:              << "1. X-Ray\n"
481:              << "2. Blood Test\n"
482:              << "3. MRI Scans\n"
483:              << "4. CT Scans\n"
484:              << "5. Physiotherapy\n"
485:              << "6. Vaccinations\n"
486:              << "7. Return to Menu\n"
487:              << "------------------------------\n"
488:              << "Please pick the service received by the patient on the " << i <<  "
```

```
    time: ";

489:         cin >> serviceTypes;
490:         system("cls");
491:
492:         while(serviceTypes < FIRST_CHOICE || serviceTypes > EXIT_DECISION)
493:         {
494:             cout << "Service Categories\n"
495:                  << "-------------------------------\n"
496:                  << "1. X-Ray\n"
497:                  << "2. Blood Test\n"
498:                  << "3. MRI Scans\n"
499:                  << "4. CT Scans\n"
500:                  << "5. Physiotherapy\n"
501:                  << "6. Vaccinations\n"
502:                  << "7. Return to Menu\n"
503:                  << "-------------------------------\n"
504:                  << "Please pick a valid option: ";
505:             cin >> serviceTypes;
506:             system("cls");
507:         }
508:
509:         switch(serviceTypes)
510:         {
511:             case FIRST_CHOICE:
512:             {
513:                 serviceFees += xRayFee;
514:                 break;
515:             }
516:             case SECOND_CHOICE:
517:             {
518:                 serviceFees += bloodTestFee;
519:                 break;
520:             }
521:             case THIRD_CHOICE:
522:             {
523:                 serviceFees += mriFee;
524:                 break;
525:             }
526:             case FOURTH_CHOICE:
527:             {
528:                 serviceFees += ctFee;
529:                 break;
530:             }
531:             case FIFTH_CHOICE:
532:             {
533:                 cout << "How many hours of physiotherapy received by the patient
    each time: ";
534:                 cin >> physioDuration;
535:                 serviceFees += (physiotherapyFee * physioDuration);
536:                 cout << endl;
537:                 break;
538:             }
539:             case SIXTH_CHOICE:
540:             {
541:                 serviceFees += vaccinationFee;
542:                 break;


543:             }
544:             default:
545:                 break;
546:         }
547:     }
548:     return serviceFees;
549: }
550:
551:
552: //Calculate and return the sum of hospital stays charges, surgery charges, pharmacy
    charges and other services charges
553: double patientBill(double hospitalStayCost, double surgeryCost, double pharmacyCost,
    double serviceCost)
```

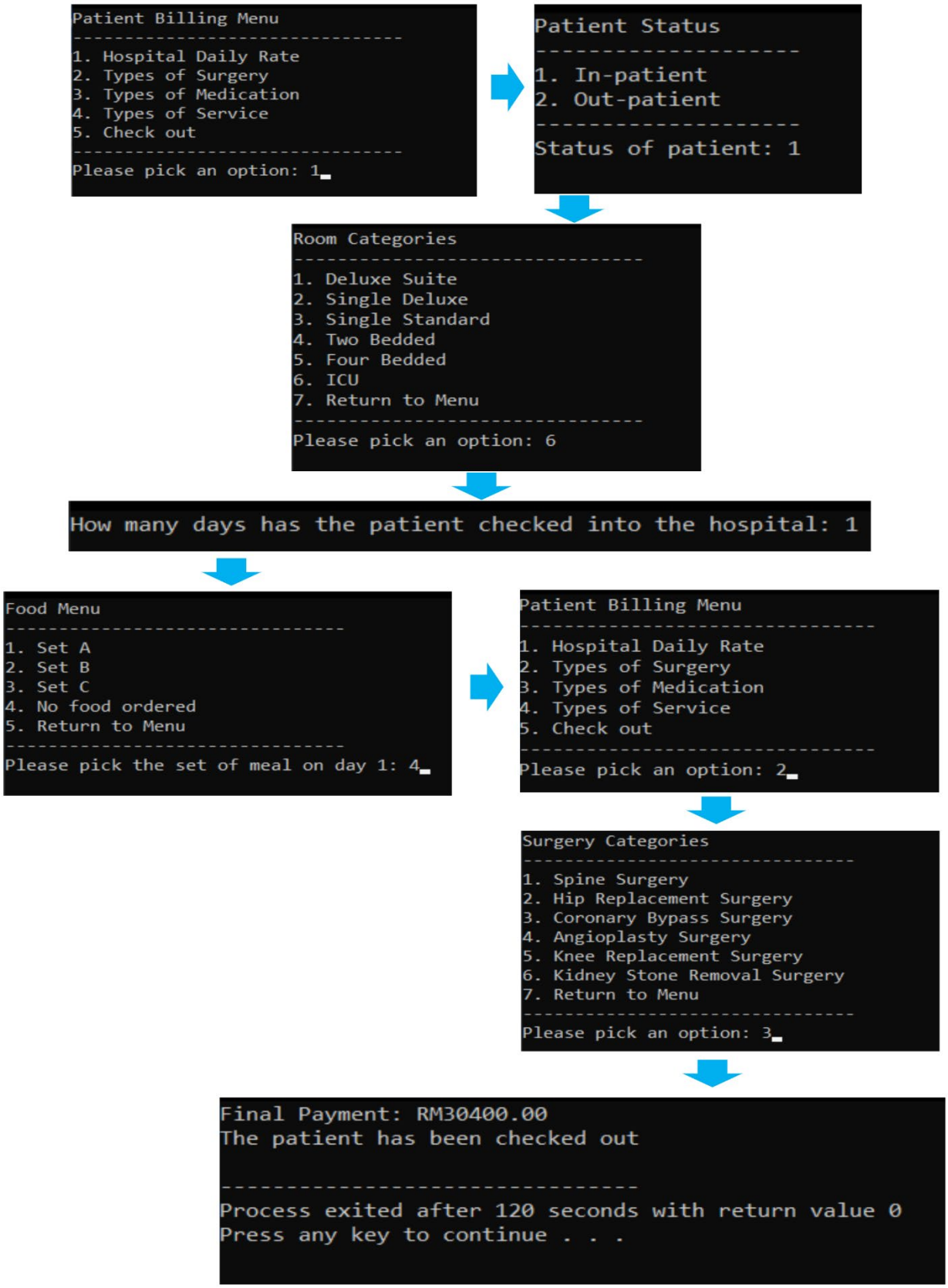
```
554: {
555:     return hospitalStayCost + surgeryCost + pharmacyCost + serviceCost;
556: }
557:
558:
559:
560: int main()
561: {
562:     int decision;
563:     const int CHECKOUT = 5;
564:
565:     double  hospitalStayCost,
566:             surgeryCost,
567:             pharmacyCost,
568:             serviceCost,
569:             totalCost;
570:
571:     do
572:     {
573:         showMenu();
574:         cout << "Please pick an option: ";
575:         cin >> decision;
576:         system("cls");
577:
578:         while(decision > CHECKOUT || decision < FIRST_CHOICE)
579:         {
580:             showMenu();
581:             cout << "Please pick a valid option: ";
582:             cin >> decision;
583:             system("cls");
584:         }
585:
586:         if(decision != CHECKOUT)
587:         {
588:             switch (decision)
589:             {
590:                 case FIRST_CHOICE:
591:                 {
592:                     hospitalStayCost = hospitalStay();
593:                     break;
594:                 }
595:                 case SECOND_CHOICE:

596:                 {
597:                     surgeryCost = surgeryCharges();
598:                     break;
599:                 }
600:                 case THIRD_CHOICE:
601:                 {
602:                     pharmacyCost = pharmacyCharges();
603:                     break;
604:                 }
605:                 case FOURTH_CHOICE:
606:                 {
607:                     serviceCost = serviceCharges();
608:                     break;
609:                 }
610:                 default:
611:                     break;
612:             }
613:         }
614:     }while(decision != CHECKOUT);
615:
616:     totalCost = patientBill(hospitalStayCost, surgeryCost, pharmacyCost,
    serviceCost);
617:     cout << fixed << setprecision(2) << "Final Payment: RM" << totalCost << endl;
    //Display the patient's bills
618:     cout << "The patient has been checked out" << endl;
619:
620:     return 0;
621: }
```

6.0) Sample Output

1)



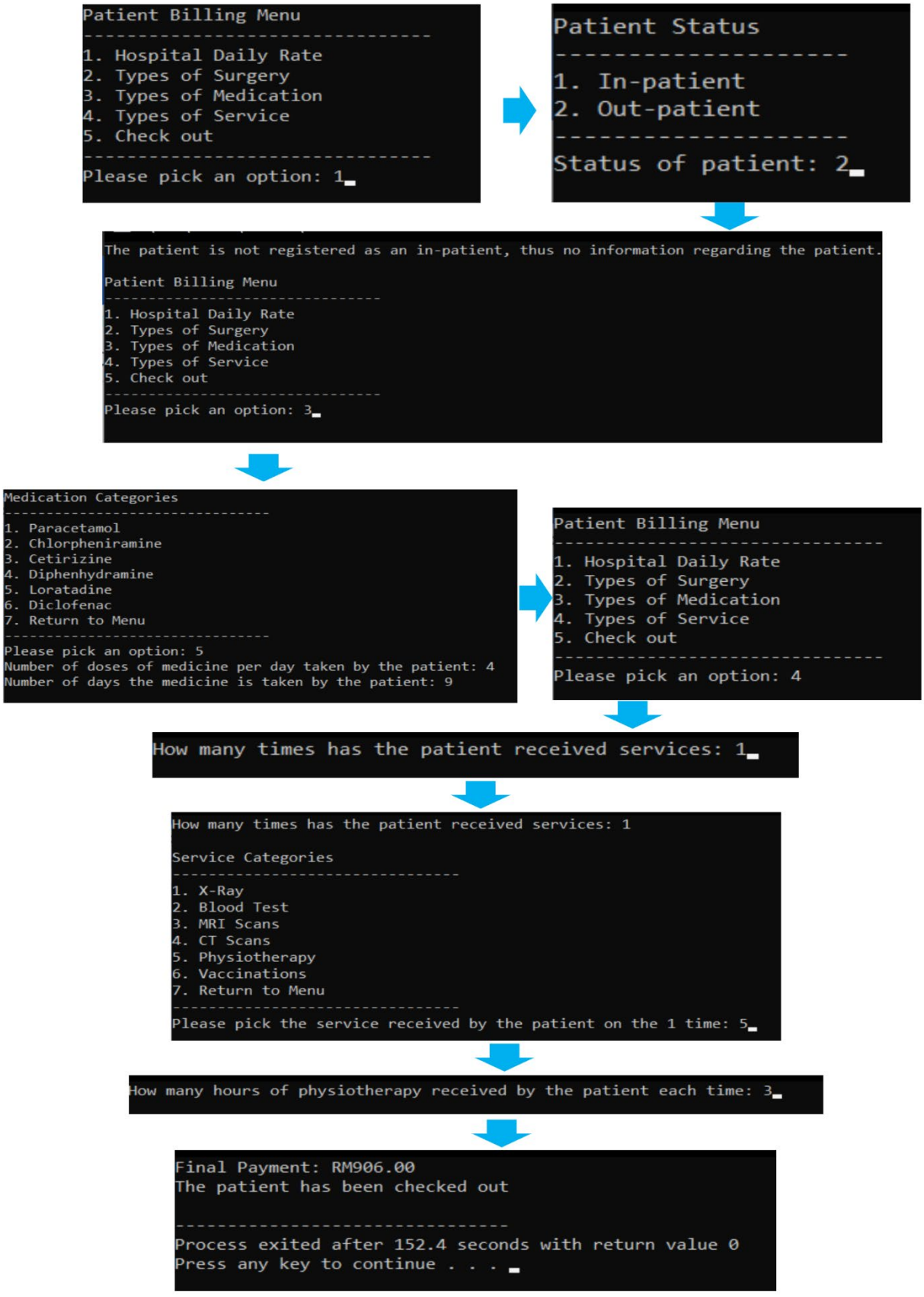
2)


Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 1
Patient Status
1. In-patient
2. Out-patient
Status of patient: 1
Room Categories
1. Deluxe Suite
2. Single Deluxe
3. Single Standard
4. Two Bedded
5. Four Bedded
6. ICU
7. Return to Menu
Please pick an option: 6
How many days has the patient checked into the hospital: 1
Food Menu
1. Set A
2. Set B
3. Set C
4. No food ordered
5. Return to Menu
Please pick the set of meal on day 1: 4
Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 2
Surgery Categories
1. Spine Surgery
2. Hip Replacement Surgery
3. Coronary Bypass Surgery
4. Angioplasty Surgery
5. Knee Replacement Surgery
6. Kidney Stone Removal Surgery
7. Return to Menu
Please pick an option: 3
Final Payment: RM30400.00
The patient has been checked out
Process exited after 120 seconds with return value 0
Press any key to continue . . .

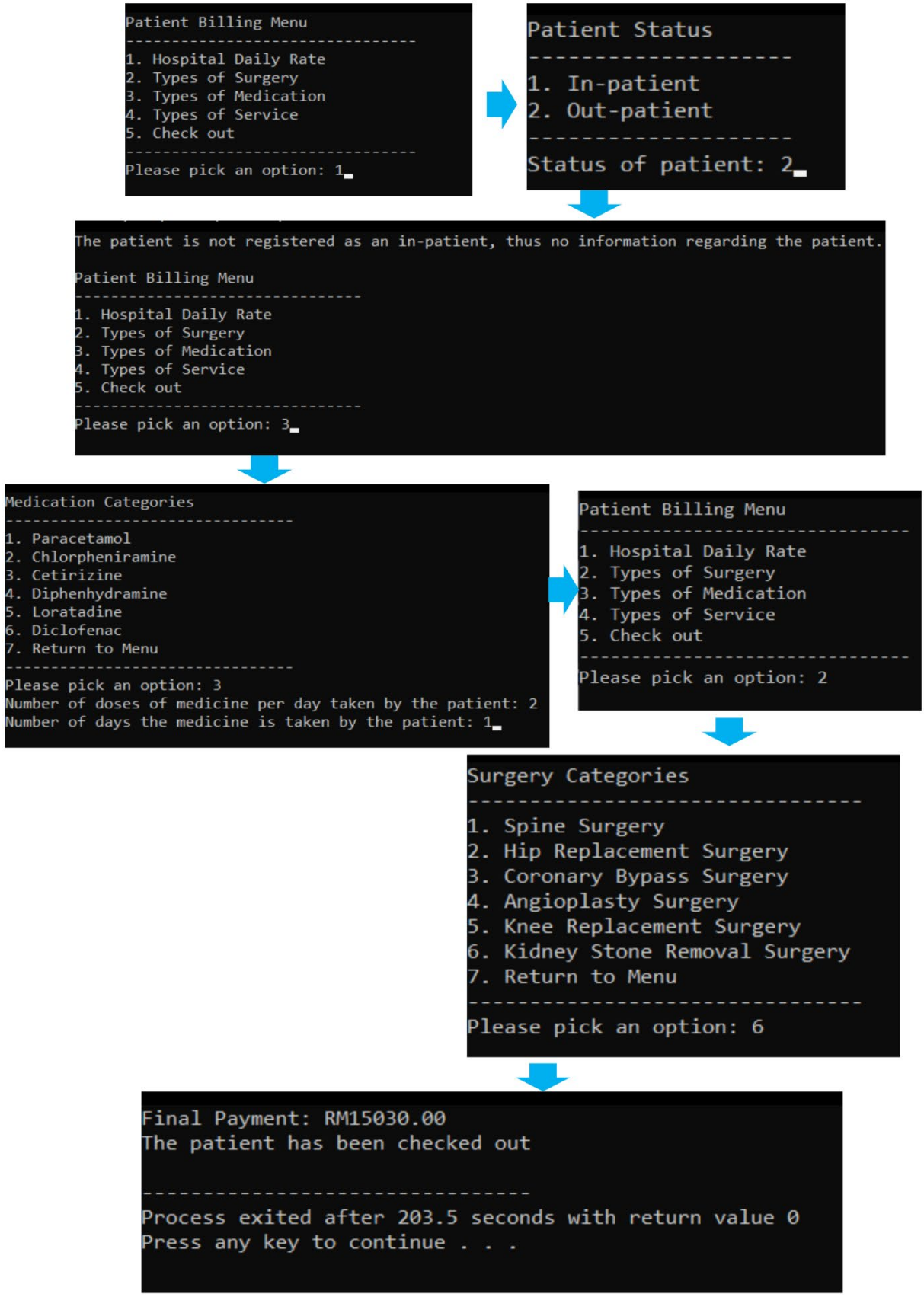
3)


Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 1
Patient Status
1. In-patient
2. Out-patient
Status of patient: 1
Room Categories
1. Deluxe Suite
2. Single Deluxe
3. Single Standard
4. Two Bedded
5. Four Bedded
6. ICU
7. Return to Menu
Please pick an option: 1
Food Menu
1. Set A
2. Set B
3. Set C
4. No food ordered
5. Return to Menu
Please pick the set of meal on day 1: 1
Food Menu
1. Set A
2. Set B
3. Set C
4. No food ordered
5. Return to Menu
Please pick the set of meal on day 2: 3
Food Menu
1. Set A
2. Set B
3. Set C
4. No food ordered
5. Return to Menu
Please pick the set of meal on day 3: 4
Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 2
Surgery Categories
1. Spine Surgery
2. Hip Replacement Surgery
3. Coronary Bypass Surgery
4. Angioplasty Surgery
5. Knee Replacement Surgery
6. Kidney Stone Removal Surgery
7. Return to Menu
Please pick an option: 1
Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 3
Medication Categories
1. Paracetamol
2. Chlorpheniramine
3. Cetirizine
4. Diphenhydramine
5. Loratadine
6. Diclofenac
7. Return to Menu
Please pick an option: 1
Number of doses of medicine per day taken by the patient: 3
Number of days the medicine is taken by the patient: 5
How many times has the patient received services: 3
Service Categories
1. X-Ray
2. Blood Test
3. MRI Scans
4. CT Scans
5. Physiotherapy
6. Vaccinations
7. Return to Menu
Please pick the service received by the patient on the 1 time: 1
Service Categories
1. X-Ray
2. Blood Test
3. MRI Scans
4. CT Scans
5. Physiotherapy
6. Vaccinations
7. Return to Menu
Please pick the service received by the patient on the 2 time: 3
Service Categories
1. X-Ray
2. Blood Test
3. MRI Scans
4. CT Scans
5. Physiotherapy
6. Vaccinations
7. Return to Menu
Please pick the service received by the patient on the 3 time: 6
Final Payment: RM58654.50
The patient has been checked out
Process exited after 528 seconds with return value 0
Press any key to continue . . .

4)


Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 1
Patient Status
1. In-patient
2. Out-patient
Status of patient: 2
The patient is not registered as an in-patient, thus no information regarding the patient.
Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 3
Medication Categories
1. Paracetamol
2. Chlorpheniramine
3. Cetirizine
4. Diphenhydramine
5. Loratadine
6. Diclofenac
7. Return to Menu
Please pick an option: 5
Number of doses of medicine per day taken by the patient: 4
Number of days the medicine is taken by the patient: 9
Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 4
How many times has the patient received services: 1
How many times has the patient received services: 1
Service Categories
1. X-Ray
2. Blood Test
3. MRI Scans
4. CT Scans
5. Physiotherapy
6. Vaccinations
7. Return to Menu
Please pick the service received by the patient on the 1 time: 5
How many hours of physiotherapy received by the patient each time: 3
Final Payment: RM906.00
The patient has been checked out
Process exited after 152.4 seconds with return value 0
Press any key to continue . . .

5)


Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 1
Patient Status
1. In-patient
2. Out-patient
Status of patient: 2
The patient is not registered as an in-patient, thus no information regarding the patient.
Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 3
Medication Categories
1. Paracetamol
2. Chlorpheniramine
3. Cetirizine
4. Diphenhydramine
5. Loratadine
6. Diclofenac
7. Return to Menu
Please pick an option: 3
Number of doses of medicine per day taken by the patient: 2
Number of days the medicine is taken by the patient: 1
Patient Billing Menu
1. Hospital Daily Rate
2. Types of Surgery
3. Types of Medication
4. Types of Service
5. Check out
Please pick an option: 2
Surgery Categories
1. Spine Surgery
2. Hip Replacement Surgery
3. Coronary Bypass Surgery
4. Angioplasty Surgery
5. Knee Replacement Surgery
6. Kidney Stone Removal Surgery
7. Return to Menu
Please pick an option: 6
Final Payment: RM15030.00
The patient has been checked out
Process exited after 203.5 seconds with return value 0
Press any key to continue . . .