

## Laboratory 8

### Implementation and Coding Standards

#### 1. Introduction and Purpose of Experiment

Students will use JavaDoc and similar standards to ensure good documentation and maintainability of code

#### 2. Aim and Objectives

**Aim:** To implement a given design with appropriate coding standards

**Objectives:** At the end of this lab, the student will be able to

- Apply industry standard coding standards
- Use automatic documentation tools
- Create maintainable code

#### 3. Experimental Procedure

- Work in teams of 4 students
- Each team should read the class diagram and identify objects, interactions and states of objects
- Each team will then split workload and develop classes individually.
- Each individual will then write their lab manual, documenting their observations

#### 4. Calculations/Computations/Algorithms

Smart Table Assistant Class:

```
1
2  /*
3   * SmartTableAssistant.java
4   *
5   * Written by : Shridhar Hegde
6   * Written for: XYZ Hotel
7   * Date      : 5th April 2018
8   * Version   : 1.0
9   *
10  * Modified by: Shivakumar M, Santosh G, Sai Bharadwaj
11  * Written for: XYZ Hotel
12  * Date      : 18th April, 2018
13  * Version   : 1.1
14  *
15  */
16
17  /*
18   * A class that represents the Smart Table Assistant. STA in short is the heart of this system as it coordinates every activity happening.
19   * It acts as an interface between user and the hotel.
20   *
21   * Version 1.1: Made interface available in 3 other languages Kannada,Tamil,Telgu
22   *
23   * @Author    Shridhar Hegde, Shivakumar M, Santoshb G, Sai Bharadwaj
24   * @Version   1.1, 18/April/2018
25   * @See      Customer, Chef, ServingTeam, Cashier, Manager
26  */
```

```

29 package smarttableassistant;
30 import java.util.*;
31 import java.util.concurrent.TimeUnit;
32
33 public class SmartTableAssistant
34 {
35     private int languageFlag, waitingTime, foodRating, customerBill=0, paymentMode;
36     private int onlinePaymentFlag = 0;
37     private String customerName, customerEmail, customerTableID;
38     private ArrayList<Integer> selectedItems = new ArrayList<Integer>();
39     private String foodFeedback, overallFeedback;
40
41     public void haltSomeTime() throws InterruptedException
42     {
43         TimeUnit.SECONDS.sleep(1);
44         System.out.print("...");
45     }
46
47     public String randomIDgenerator()
48     {
49         char[] chars = "abcdefghijklmnopqrstuvwxyz0123456789".toCharArray();
50         StringBuilder sb = new StringBuilder(20);
51         Random random = new Random();
52         for (int i = 0; i < 8; i++)
53         {
54             char c = chars[random.nextInt(chars.length)];
55             sb.append(c);
56         }
57         System.out.print(sb);
58         return(sb.toString());
59     }
60 }

```

```

63 //Changes the interface language as per customer's needs
64 public void changeLanguage(Customer customer) throws InputMismatchException
65 {
66     Scanner input = new Scanner(System.in);
67     boolean validInput = false;
68
69     //Generate unique table ID
70     customerTableID = randomIDgenerator();
71     customer.generateTableID(customerTableID);
72
73     //Ask customer to choose language
74     System.out.println("Choose any one of the language:");
75     System.out.println("1. English\n2. Kannada\n3. Telugu\n5. Tamil");
76     System.out.print("\t$ ");
77
78     //To take the language preference input from the user
79     while(!validInput)
80     {
81         try
82         {
83             //Take language preference from customer class
84             this.languageFlag = customer.chooseLanguage();
85
86             if(this.languageFlag<1 || this.languageFlag>5)
87             {
88                 System.out.println("\tWrong input. Enter again !");
89                 System.out.print("\t$ ");
90             }
91             else
92             {
93                 validInput = true;
94             }
95         }
96         catch(InputMismatchException e)
97         {
98             System.out.print("\tEnter an integer !! Please try again !\n\t$ ");
99         }
100     }
101 }

```

```

143 public final void menuInKannada(Customer customer, Chef chef,ServingTeam servingTeam)throws InterruptedException
144 {
145     Scanner input = new Scanner(System.in);
146     char choice;
147
148     System.out.println("-----");
149
150     //Take user details
151     System.out.print("Nimma vivaragalannu namudisi:\n\t*Name: ");
152     this.customerName = customer.inputName(input);
153     System.out.print("\t*E-mail: ");
154     this.customerEmail = customer.inputEmail(input);
155
156     //Display menu in english
157     System.out.println("Kelagina menuvininda nimma adeshavannu ayke madi: \n\t(Suchane: Nimage yavude item estu beko ayke madi.");
158     System.out.println("\t\t\t\t\tNimage onde item jasti bekiddalli astu sala aa number annu enter madi. Mattu konege \"0\" annu enter madi");
159     System.out.println("1.Bisibele Bath - ?50\t\t2.Vangi Bath - ?70\n3.Rice Bath - ?50\t\t4.Pongal - ?80\n5.Shavige Bath - ?60\t\t6.Idli-Vada - ?60");
160     System.out.print("\t$t ");
161     while(true)
162     {
163         selectedItems = customer.chooseItems(input);
164
165         waitingTime = chef.takeOrder(selectedItems);
166
167         System.out.println("\n\tDAYAVITTU " +waitingTime+" SECONDS KAYIRI\n");
168         haltSomeTime(); haltSomeTime(); haltSomeTime(); haltSomeTime(); haltSomeTime(); haltSomeTime();
169
170         //Remind chef to notify serving team as to serve the food to customers
171         chef.notifyServingTeam(customerTableID, customer, servingTeam);
172
173         System.out.print("NIVU MATTE ORDER BAYASUTTIRA ? [Y/N]: ");
174         choice = input.next().charAt(0);
175         Character ch = Character.toUpperCase(choice);
176
177         if(ch.equals('N'))
178             break;
179     }
180 }

```

4

```

262     public void placingOrder(Customer customer, Chef chef, ServingTeam servingTeam) throws InterruptedException
263     {
264         switch(this.languageFlag)
265         {
266             case 1:
267                 this.menuInEnglish(customer, chef, servingTeam);
268                 break;
269             case 2:
270                 this.menuInKannada(customer, chef, servingTeam);
271                 break;
272             case 3:
273                 this.menuInTelugu(customer, chef, servingTeam);
274                 break;
275             case 4:
276                 this.menuInTamil(customer, chef, servingTeam);
277                 break;
278         }
279     }
280
281     public void feedback(Chef chef, Customer customer, Manager manager)
282     {
283         Scanner input = new Scanner(System.in);
284         switch(this.languageFlag)
285         {
286             case 1:
287                 this.feedbackInEnglish(input, chef, customer, manager);
288                 break;
289             case 2:
290                 this.feedbackInKannada(input, chef, customer, manager);
291                 break;
292             case 3:
293                 this.feedbackInTelugu(input, chef, customer, manager);
294                 break;
295             case 4:
296                 this.feedbackInTamil(input, chef, customer, manager);
297                 break;
298         }
299     }
300

```

```

301     public void feedbackInEnglish(Scanner input, Chef chef, Customer customer, Manager manager)
302     {
303         System.out.println("-----");
304         System.out.println("Dear customer, thank you for choosing our hotel.\nIn order to upgrade our services, we expect feedbacks from our customers.");
305
306         //Take food rating
307         System.out.print("Please rate the food from 1 to 10: ");
308         foodRating = Integer.parseInt(customer.feedback(1, input));
309
310         //Take food feedback
311         System.out.print("Please your feedback regarding the food quality here:\n\t$ ");
312         this.foodFeedback = customer.feedback(2, input);
313
314         //Take overall feedback
315         System.out.print("Please leave your feedback for the overall experience of the hotel:\n\t$ ");
316         this.overallFeedback = customer.feedback(3, input);
317
318         //Forwarding food feedback to chef and both overall & food feedback to manager
319         chef.receiveCustomerFeedback(foodFeedback);
320         manager.receiveCustomerFeedback(foodRating, foodFeedback, overallFeedback, chef);
321
322         System.out.println("\n Thank you for the precious feedback. Now you can move on for payment. ");
323         System.out.println("-----");
324     }
325

```

```

329 public void feedbackInKannada(Scanner input, Chef chef, Customer customer, Manager manager)
330 {
331     System.out.println("-----");
332
333     System.out.println("Priya grahaka, namma hotel annu ayke madidakkagi dhanyavadagalu.\n
334     Namma sevegalannu upgrade madalu navu namma grahakara pratikriyeyannu niriksisutteve.");
335
336     //Take food rating
337     System.out.print("Dayavittu aharavannu 1 rinda 10 ravarege rate madi: ");
338     foodRating = Integer.parseInt(customer.feedback(1, input));
339
340     //Take food feedback
341     System.out.print("Illi aharada gunamattada bagge nimma pratikriyeyannu irisi:\n\t$ ");
342     this.foodFeedback = customer.feedback(2, input);
343
344     //Take overall feedback
345     System.out.print("Sthalada ottare anubavada bagge nimma pratikriyeyannu bittukodi:\n\t$ ");
346     this.overallFeedback = customer.feedback(3, input);
347
348     //Forwarding food feedback to chef and both overall & food feedback to manager
349     chef.receiveCustomerFeedback(foodFeedback);
350     manager.receiveCustomerFeedback(foodRating, foodFeedback, overallFeedback, chef);
351
352
353     System.out.println("\n Nimma amulya pratikriyegagi dhanyavadagalu. Iga nivu pavatigagi chalisabahudu. ");
354     System.out.println("-----");
355 }

```

```

357 public void feedbackInTelugu(Scanner input, Chef chef, Customer customer, Manager manager)
358 {
359     System.out.println("-----");
360
361     System.out.println("Priyamaina customer, meeru maa hotelnu enchukunnanduku dhanyavaadalu.\n
362     Ma sevalanu upgrade cheyyadaniki, memu ma kastamarla nundi abhiprayalanu asisthunnamu.");
363
364     //Take food rating
365     System.out.print("Dayachesi 1 nundi 10 varaku aaharanni rate cheyyandi: ");
366     foodRating = Integer.parseInt(customer.feedback(1, input));
367
368     //Take food feedback
369     System.out.print("Ikkadi aahara nanyathaku sambhandhinchu mii feedback ivvandi:\n\t$ ");
370     this.foodFeedback = customer.feedback(2, input);
371
372     //Take overall feedback
373     System.out.print("Dayachesi hotel yokka motham anubhavaniki mi abhipraayanni teliya cheyandi:\n\t$ ");
374     this.overallFeedback = customer.feedback(3, input);
375
376     //Forwarding food feedback to chef and both overall & food feedback to manager
377     chef.receiveCustomerFeedback(foodFeedback);
378     manager.receiveCustomerFeedback(foodRating, foodFeedback, overallFeedback, chef);
379
380
381     System.out.println("\n Viluvaina feedback ku dhanyavaadhalu. Ippudu miru chellimpu kosam vellachu. ");
382     System.out.println("-----");
383 }

```

```

385 public void feedbackInTamil(Scanner input, Chef chef, Customer customer, Manager manager)
386 {
387     System.out.println("-----");
388
389     System.out.println("Vitikkai aalar, yengal hotelai thernteduthadhukku nandri.\n
390     Enkal sevaigalai mempatuthavadhukku, vatikkaiaalarkaliranthu karuthukkalai ethirpaakirom.");
391
392     //Take food rating
393     System.out.print("1 mudhal 10 varai unavukku madipengalai kodukavaum: ");
394     foodRating = Integer.parseInt(customer.feedback(1, input));
395
396     //Take food feedback
397     System.out.print("Inge unavu tharam pathiya unkal karutthakali aluthavum:\n\t$ ");
398     this.foodFeedback = customer.feedback(2, input);
399
400     //Take overall feedback
401     System.out.print("Indha hotelin ottumotha tharam pathiya unkal karuthukalai aluthavum\n\t$ ");
402     this.overallFeedback = customer.feedback(3, input);
403
404     //Forwarding food feedback to chef and both overall & food feedback to manager
405     chef.receiveCustomerFeedback(foodFeedback);
406     manager.receiveCustomerFeedback(foodRating, foodFeedback, overallFeedback, chef);
407
408
409     System.out.println("\n Vilaimathipana karuthukalukku nandri. Ippothu neenkal kattanam sellutha mudiyum ");
410     System.out.println("-----");
411 }

```

```

414     public void billPayment(Customer customer, Cashier cashier)
415     {
416         Scanner input = new Scanner(System.in);
417         switch(this.languageFlag)
418         {
419             case 1:
420                 this.billPaymentInEnglish(customer,input,cashier);
421                 break;
422             case 2:
423                 this.billPaymentInKannada(customer,input,cashier);
424                 break;
425             case 3:
426                 this.billPaymentInTelugu(customer,input,cashier);
427                 break;
428             case 4:
429                 this.billPaymentInTamil(customer,input,cashier);
430                 break;
431         }
432     }

```

```

434     public void billPaymentInEnglish(Customer customer,Scanner input,Cashier cashier)
435     {
436         System.out.println("\t\tBILL PAYMENT ");
437         for(int i=0; i < selectedItems.size(); i++)
438         {
439             switch(this.selectedItems.get(i))
440             {
441                 case 1: this.customerBill+=50; break;
442                 case 2: this.customerBill+=70; break;
443                 case 3: this.customerBill+=50; break;
444                 case 4: this.customerBill+=80; break;
445                 case 5: this.customerBill+=60; break;
446                 case 6: this.customerBill+=60; break;
447             }
448         }
449         System.out.println("You bill is ? "+customerBill+".00.");
450         System.out.println("How would you like to pay the bill ? Choose any one option.");
451         System.out.println("\t1. Debit/Credit Card\n\t2. Cash Payment");
452         this.paymentMode = Integer.parseInt(customer.payingBill(1, input));
453         if(this.paymentMode==1)
454         {
455             System.out.println(" WE ARE TRANSFERRING YOU TO BANK PORTAL.\nNote: We do not store your payment information");
456             System.out.println("-----");
457             while(true)
458             {
459                 if(onlinePaymentFlag == 1)
460                     break;
461                 System.out.print("Enter card number: ");
462                 String dummy = customer.payingBill(2, input);
463                 System.out.print("Enter pin: ");
464                 dummy = customer.payingBill(3, input);
465                 System.out.print("Enter the amount: ");
466                 dummy = customer.payingBill(4, input).toString();
467                 //Checking if the payed amount is less than the bill amount, we have ask customer to pay again
468                 if(Integer.parseInt(dummy) < this.customerBill)
469                     onlinePaymentFlag=0;
470                 else

```

```

471                     onlinePaymentFlag = 1;
472                 System.out.println("-----");
473                 if(this.onlinePaymentFlag==1)
474                     System.out.println(".....\n\tPayment Successful.");
475                 else
476                     System.out.println("Transaction error ! Try again.");
477             }
478         }
479         else
480         {
481             System.out.println("Please pay the bill in cash to the cashier &\n\t don't forget to collect the acknowledgment.");
482             cashier.customerPayingCash(customerTableID,customerBill);
483         }
484     }

```

```

486 public void billPaymentInKannada(Customer customer,Scanner input,Cashier cashier)
487 {
488     System.out.println("\t\tBILL PAAVATI ");
489     for(int i=0; i < selectedItems.size(); i++)
490     {
491         switch(this.selectedItems.get(i))
492         {
493             case 1: this.customerBill+=50; break;
494             case 2: this.customerBill+=70; break;
495             case 3: this.customerBill+=50; break;
496             case 4: this.customerBill+=80; break;
497             case 5: this.customerBill+=60; break;
498             case 6: this.customerBill+=60; break;
499         }
500     }
501     System.out.println("Nimma bill ? "+customerBill+".00.");
502     System.out.println("Nivu bill annu hege paavatisuttiri ? Yavude ondu aykeyannu arisi.");
503     System.out.println("\t1. Debit/Credit Card\t2. Cash Payment");
504     this.paymentMode = Integer.parseInt(customer.payingBill(1, input));
505
506     if(this.paymentMode==1)
507     {
508         System.out.println(" NAVU NIMMANNU BANK PORTALGE VARGAYISUTTIDDEVE.\nSuchane: Nimma paavatiya mahitiyannu navu sangrahisuvudilla");
509
510         System.out.println("-----");
511
512         while(true)
513         {
514             if(onlinePaymentFlag == 1)
515                 break;
516             System.out.print("Card sankhye namudisi: ");
517             String dummy = customer.payingBill(2, input);
518             System.out.print("Pin annu namudisi: ");
519             dummy = customer.payingBill(3, input);
520             System.out.print("Mottavannu namudisi: ");
521             dummy = customer.payingBill(4, input).toString();
522
523             //Checking if the payed amount is less than the bull amount, we have ask customer to pay again
524             if(Integer.parseInt(dummy) < this.customerBill)
525                 onlinePaymentFlag=0;
526             else
527                 onlinePaymentFlag = 1;
528
529             System.out.println("-----");
530
531             if(this.onlinePaymentFlag==1)
532                 System.out.println(".....\n\tPaavati yavasviyagide.");
533             else
534                 System.out.println("Transaction dosha ! Mattome prayatnisi.");
535         }
536     }
537     else
538     {
539         System.out.println("Paavati annu cashierge balasi &\n\t Acknowledgment tagedukollalu marebedi.");
540
541         cashier.customerPayingCash(customerTableID,customerBill);
542     }
543 }
544 }
545 }

```



```

547 public void billPaymentInTelugu(Customer customer, Scanner input, Cashier cashier)
548 {
549     System.out.println("\t\tBILLU CHELLIMPU ");
550     for(int i=0; i < selectedItems.size(); i++)
551     {
552         switch(this.selectedItems.get(i))
553         {
554             case 1: this.customerBill+=50; break;
555             case 2: this.customerBill+=70; break;
556             case 3: this.customerBill+=50; break;
557             case 4: this.customerBill+=80; break;
558             case 5: this.customerBill+=60; break;
559             case 6: this.customerBill+=60; break;
560         }
561     }
562     System.out.println("Mi billu ? "+customerBill+".00.");
563     System.out.println("Miru billunu ela chellinchalanukuntunnaru ? Edaina oka empikanu Enchukondi.");
564     System.out.println("\t1. Debit/Credit Card\n\t2. Cash Payment");
565     this.paymentMode = Integer.parseInt(customer.payingBill(1, input));
566
567     if(this.paymentMode==1)
568     {
569         System.out.println("MEMU BANK PORTAL KU BADILI CHESTUNNAMU.\nGamanika: memu mi chellimpu samaacharanni nilva cheyyalemu");
570
571         System.out.println("-----");
572
573         while(true)
574         {
575             if(onlinePaymentFlag == 1)
576                 break;
577             System.out.print("Card sankhyanu namodu cheyyandi: ");
578             String dummy = customer.payingBill(2, input);
579             System.out.print("Pin namodu cheyyandi: ");
580             dummy = customer.payingBill(3, input);
581             System.out.print("Mottanni namodu cheyyandi: ");
582             dummy = customer.payingBill(4, input).toString();
583
584             //Checking if the payed amount is less than the bull amount, we have ask customer to pay again
585             if(Integer.parseInt(dummy) < this.customerBill)
586                 onlinePaymentFlag=0;
587             else
588                 onlinePaymentFlag = 1;
589
590             System.out.println("-----");
591
592             if(this.onlinePaymentFlag==1)
593                 System.out.println(".....\n\tChellimpu vijayavantamainadi.");
594             else
595                 System.out.println("Lava devilalo lopam ! Malli prayatninchandi.");
596         }
597     }
598     else
599     {
600         System.out.println("Dayachesi cashu nagadulo billunu chellinchandi Mariyu\n\t rasidu sekarinchadam marachipovaddu.");
601
602         cashier.customerPayingCash(customerTableID, customerBill);
603
604     }
605 }
606
607 }

```

```

609 public void billPaymentIamail(Customer customer, Scanner input,Cashier cashier)
610 {
611     System.out.println("\t\tKattanam selutahavum: ");
612     for(int i=0; i < selectedItems.size(); i++)
613     {
614         switch(this.selectedItems.get(i))
615         {
616             case 1: this.customerBill+=50; break;
617             case 2: this.customerBill+=70; break;
618             case 3: this.customerBill+=50; break;
619             case 4: this.customerBill+=80; break;
620             case 5: this.customerBill+=60; break;
621             case 6: this.customerBill+=60; break;
622         }
623     }
624     System.out.println("Unkal Raceethu: ? "+customerBill+".00.");
625     System.out.println("Eppadi Kattanam selutha virumarirkal? Enta oru virupattaiyum thervu seiyyavum.");
626     System.out.println("\t1. Debit/Credit Card\n\t2. Cash Payment");
627     this.paymentMode = Integer.parseInt(customer.payingBill(1, input));
628
629     if(this.paymentMode==1)
630     {
631         System.out.println("Nankal unkalai vanki portilluku alaithu chelkirom.\nKurippu: Nankal kattanathu thakavalai semikkamaatom");
632
633         System.out.println("-----");
634
635         while(true)
636         {
637             if(onlinePaymentFlag == 1)
638                 break;
639             System.out.print("ATM attai en aluthavum: ");
640             String dummy = customer.payingBill(2, input);
641             System.out.print("ATM pin en aluthavum: ");
642             dummy = customer.payingBill(3, input);
643             System.out.print("Panam yevalavu endru aluthavum: ");
644             dummy = customer.payingBill(4, input).toString();
645
646             //Checking if the payed amount is less than the bull amount, we have ask customer to pay again
647             if(Integer.parseInt(dummy) < this.customerBill)
648                 onlinePaymentFlag=0;
649             else
650                 onlinePaymentFlag = 1;
651
652             System.out.println("-----");
653
654             |
655             if(this.onlinePaymentFlag==1)
656                 System.out.println(".....\n\tKattanam vetrikanagama mudindhadhu.");
657             else
658                 System.out.println("Kattanam mudiyaivillai! Marubadi muyarchikkavum.");
659         }
660     }
661     else
662     {
663         System.out.println("Cashieridam panathai seluthavum &\n\t Oppukai cekarikka marakadirkal");
664
665         cashier.customerPayingCash(customerTableID,customerBill);
666
667     }
668 }
669
670 public static void main(String[] args) throws InterruptedException
671 {
672
673     System.out.println("=====");
674     System.out.println("                                     \n" +
675 "    _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/__\n" +
676 "   /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/\n" +
677 "  /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/\n" +
678 " /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/\n" +
679 "/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/\n" +
680 "_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/\n" +
681 "_____\n");
682     System.out.println("=====");
683
684     SmartTableAssistant assisant = new SmartTableAssistant();
685
686     //Change interface language when the assistant starts
687     Customer customer = new Customer();
688     assisant.changeLanguage(customer);
689
690     //Place the order and
691     //Pass the order to chef
692     Chef chef = new Chef();
693     ServingTeam servingTeam = new ServingTeam();
694     assisant.placingOrder(customer, chef,servingTeam);
695
696

```

```

696
697 //To ask for feedback
698 Manager manager = new Manager();
699 assisant.feedback(chef,customer,manager);
700
701 //Bill Payment
702 Cashier cashier = new Cashier();
703 assisant.billPayment(customer,cashier);
704
705 System.out.println("We are happy to serve you.");
706
707 System.out.print("
708
709
710
711
712
713
714
715
716 }
717
718 }
719

```

Customer Class:

```

2 /*
3  * Customer.java
4  *
5  * Written by : Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
6  * Written for: XYZ Hotel
7  * Date      : 5th April 2018
8  * Version   : 1.0
9  *
10 */
11
12 /*
13  * This is a class that represents the customer. All the inputs given by the customer are handled by this class.
14  *
15  * Version 1.0
16  *
17  * @Author    Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
18  * @Version   1.0, 5/April/2018
19  * @See       Customer, Chef, Serving Team, Cashier, Manager
20  */
21
22 package smarttableassistant;
23

```

```

29 public class Customer
30 {
31     public boolean foodServed = false;
32     private String customerName,customerEmail, customerTableID,foodFeedback, overallFeedback, cardNumber, pin, paymentAmount;
33     private int languageFlag, itemCount=0, foodRating, paymentMode;
34     private ArrayList<Integer> selectedItems = new ArrayList<Integer>();
35
36     public void generateTableID(String CustomerTableID)
37     {
38         this.customerTableID = customerTableID;
39     }
40
41     public String inputName(Scanner input) //Takes customer name input
42     {
43         this.customerName = input.next();
44         return(this.customerName);
45     }
46
47     public String inputEmail(Scanner input) //Takes customer email input
48     {
49         this.customerEmail = input.next();
50         return(this.customerEmail);
51     }
52
53     public ArrayList<Integer> chooseItems(Scanner input) // Returns the items selected by the user
54     {
55         int choice;
56         while(true)
57         {
58             if(itemCount > 7) break;
59             choice = input.nextInt();
60             if(choice == 0) break;
61             else
62             {
63                 selectedItems.add(choice);
64                 itemCount++;
65             }
66         }
67         return(selectedItems);
68     }
69 }

```

```

70     public int chooseLanguage()           //User can choose any of the offered language
71     {
72         //Customer is storing the language preference
73         Scanner input = new Scanner(System.in);
74         this.languageFlag = input.nextInt();
75         return(this.languageFlag);
76     }
77
78     public void operateTableAssistant()
79     {
80     }
81
82
83     public String payingBill(int choice, Scanner input)
84     {
85         switch(choice)
86         {
87             case 1:
88                 this.paymentMode = input.nextInt();
89                 return(Integer.toString(this.paymentMode));
90             case 2:
91                 this.cardNumber = input.next();
92                 return("");
93             case 3:
94                 pin = input.next();
95                 return("");
96             case 4:
97                 this.paymentAmount = input.next();
98                 return(paymentAmount);
99         }
100         return("0");
101     }
102 }
103

```

```

107     public String feedback(int choice, Scanner input)
108     {
109         switch(choice)
110         {
111             case 1:
112                 while(true)
113                 {
114                     this.foodRating = input.nextInt();
115                     if(this.foodRating > 10 || this.foodRating < 1)
116                         System.out.print("Oops! Enter again: ");
117                     else
118                         return(Integer.toString(this.foodRating));
119                 }
120             case 2:
121                 this.foodFeedback = input.next();
122                 return(this.foodFeedback);
123             case 3:
124                 this.overallFeedback = input.next();
125                 return(this.overallFeedback);
126         }
127         return(" ");
128     }
129 }
130
131 }
132

```

Chef class:

```

1
2  /*
3   * Chef.java
4   *
5   * Written by : Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
6   * Written for: XYZ Hotel
7   * Date      : 5th April 2018
8   * Version   : 1.0
9   *
10  */
11
12  /*
13   This is a class that represents the chef in teh system. Chef is reposible for taking orders from the user.
14   Then he/she calculates the waiting time for the customer. After the food is prepared they notify the serving team to serve the customers.
15
16   Version 1.0
17
18   @Author    Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
19   @Version   1.0, 5/April/2018
20   @See      Customer, Chef, Serving Team, Manager
21  */
22
23  package smarttableassistant;
24  import java.util.*;
25

```

```

26  public class Chef
27  {
28      private ArrayList<Integer> customerOrder = new ArrayList<Integer>();
29      private String customerFeedback, managerRemarks;
30
31
32      public int takeOrder(ArrayList<Integer> selectedItems)
33      {
34          this.customerOrder = selectedItems;
35          Random randomNumberGenerator = new Random();
36          return(randomNumberGenerator.nextInt((1200 - 300) + 1) + 300);
37      }
38
39      public void notifyServingTeam(String customerTableID, Customer customer, ServingTeam servingTeam) throws InterruptedException
40      {
41
42          servingTeam.serveFood(customerTableID, customer, customerOrder);
43      }
44
45      public void receiveCustomerFeedback(String customerFeedback)
46      {
47          this.customerFeedback = customerFeedback;
48      }
49
50
51      public void putManagerRemarks(String managerRemarks)
52      {
53          this.managerRemarks = managerRemarks;
54      }
55
56  }
57

```

### Serving Team:

```

4   *
5   * Written by : Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
6   * Written for: XYZ Hotel
7   * Date      : 5th April 2018
8   * Version   : 1.0
9   *
10  */
11
12  /*
13   A class that represents the serving team in the hotel. It takes notification from the chef to deliver food to a
14   particular table id.
15
16   Version 1.0
17
18   @Author    Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
19   @Version   1.0, 15/April/2018
20   @See      Customer, Chef
21  */
22
23  package smarttableassistant;
24  import java.util.*;
25  import java.util.concurrent.TimeUnit;
26

```

```

27 public class ServingTeam
28 {
29     private String customerTableID;
30     private boolean foodServed = false;
31     private ArrayList<Integer> itemsToServe = new ArrayList<Integer>();
32
33     public void serveFood(String customerTableID, Customer customer, ArrayList<Integer> itemsToServe) throws InterruptedException
34     {
35         this.customerTableID = customerTableID;
36         this.itemsToServe = itemsToServe;
37
38         //Serves food to customer by changing their food served flag
39         customer.foodServed = true;
40
41         System.out.println(".....\nServing table id "+this.customerTableID+" with items:");
42         for(int i=0; i < this.itemsToServe.size(); i++)
43         {
44             switch(this.itemsToServe.get(i))
45             {
46                 case 1: System.out.println("\t\tBisibele Bath"); break;
47                 case 2: System.out.println("\t\tVangi Bath"); break;
48                 case 3: System.out.println("\t\tRice Bath"); break;
49                 case 4: System.out.println("\t\tPongal"); break;
50                 case 5: System.out.println("\t\tShavige Bath"); break;
51                 case 6: System.out.println("\t\tIdli-Vada"); break;
52             }
53         }
54         System.out.println("\tCUSTOMER STARTED EATING");
55         while(customerIsEating()); while(customerIsEating());
56         while(customerIsEating()); while(customerIsEating());
57         while(customerIsEating()); while(customerIsEating());
58         while(customerIsEating()); while(customerIsEating());
59         System.out.print("\n");
60     }
61
62     public void waitWhileCustomerIsEating() throws InterruptedException
63     {
64         TimeUnit.SECONDS.sleep(1);
65         System.out.print(".....");
66     }
67 }

```

Cashier:

```

1
2 /*
3  * Manager.java
4  *
5  * Written by : Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
6  * Written for: XYZ Hotel
7  * Date      : 5th April 2018
8  * Version   : 1.0
9  *
10 */
11
12 /*
13  * This is a class that represents the manager of the hotel. Manager is responsible for improving the service of the hotel.
14  * He also receives the feedback given by the users.
15  *
16  * Version 1.0
17  *
18  * @Author    Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
19  * @Version   1.0, 5/April/2018
20  * @See      Customer, Chef
21  */
22
23 package smarttableassistant;
24
25 public class Cashier
26 {
27     private String tableID;
28     private int amount;
29     public void customerPayingCash(String tableID, int amount)
30     {
31         this.tableID = tableID;
32         this.amount = amount;
33     }
34 }
35

```

Manager:

```
1
2
3  /*
4   * Manager.java
5   *
6   * Written by : Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
7   * Written for: XYZ Hotel
8   * Date      : 5th April 2018
9   * Version   : 1.0
10  */
11
12  /*
13   * This is a class that represents the manager of the hotel. Manager is responsible for improvising the service of the hotel.
14   * He also receives the feedback given by the users.
15   *
16   * Version 1.0
17   *
18   * @Author    Shridhar Hegde, Shivakumar M, Santosh G, Sai Bharadwaj
19   * @Version   1.0, 5/April/2018
20   * @See      Customer, Chef
21  */
22
23  package smarttableassistant;
24
```

```
25
26  public class Manager
27  {
28      private int foodRating;
29      String username, password, foodFeedback, overallFeedback;
30
31      public void receiveCustomerFeedback(int foodRating, String foodFeedback, String overallFeedback, Chef chef)
32      {
33          this.foodRating = foodRating;
34          this.foodFeedback = foodFeedback;
35          this.overallFeedback = overallFeedback;
36
37          this.improvise(chef);
38      }
39
40      public void improvise(Chef chef)
41      {
42          if(foodRating <= 5)
43              chef.putManagerRemarks("You have to talk with me regarding the improvement of food quality");
44          else
45              chef.putManagerRemarks("Well done");
46      }
47  }
48
49
```

## 5. Presentation of Results

The output for the code is:

```

c2anew0uChoose any one of the language:
1. English
2. Kannada
3. Telugu
5. Tamil
$ 2
-----
Nimma vivaragalannu namudisi:
*Name: Erlich
*E-mail: erlichbachman@gmail.com
Kelagina menuvininda nimma adeshavannu ayke madi:
(Suchane: Nimage yavude item estu beko ayke madi.
Nimage onde item jasti bekiddalli astu sala aa number annu enter madi. Mattu konege "0" annu enter madi)
1.Bisibele Bath - ?50      2.Vangi Bath - ?70
3.Rice Bath - ?50          4.Pongal - ?80
5.Shavige Bath - ?60       6.Idli-Vada - ?60
$ 1 1 2 3 5 6 6 0

DAYAVITTU 342 SECONDS KAYIRI
-----
Serving table id c2anew0u with items:
Bisibele Bath
Bisibele Bath
Vangi Bath
Rice Bath
Shavige Bath
Idli-Vada
-----
Idli-Vada
CUSTOMER STARTED EATING
-----
NIVU MATTE ORDER BAYASUTTIRA ? [Y/N]: n
-----
Priya grahaka, namma hotel annu ayke madidakkagi dhanyavadagalu.
Namma sevegalannu upgrade madalu navu namma grahakara pratikriyeyannu niriksisutteve.
Dayavittu aharavannu 1 rinda 10 ravarege rate madi: 5
Illi aharada gunamattada bagge nimma pratikriyeyannu irisi:
$ Chennagittu
Sthalada ottare anubavada bagge nimma pratikriyeyannu bittukodi:
$ Bahala chennagittu

Nimma amulya pratikriyegagi dhanyavadagalu. Iga nivu pavatigagi chalisabahudu.
-----
BILL PAAVATI
Nimma bill ? 400.00.
Nivu bill annu hege paavatisuttiri ? Yavude ondu aykeyannu arisi.
1. Debit/Credit Card
2. Cash Payment
1
NAVU NIMMANNU BANK PORTALGE VARGAYISUTTIDDEVE.
Suchane: Nimma paavatiya mahitiyannu navu sangrahisuvudilla
-----
Card sankhye namudisi: 012345678912345
Pin annnu namudisi: 0000
Mottavannu namudisi: 400
-----
-----
Paavati yasasviyagide.
We are happy to serve you.

```

## 6. Analysis and Discussions



In the current laboratory, the system designed using UML in the previous lab sessions are **implemented using Java**. Some of the **java coding conventions** are also followed while developing the system so that anyone who reads the source code clearly understand what is the functionality of the each functions. The conventions followed are:

- The name of variables and functions are related to their significance in the code
- A variable name starts with a small letter and if there are more than one words in the name, first letter of the subsequent words are capitalised. The same is followed for functions.
- The name of the class starts with a capital letter and first letter of the subsequent words are capitalised.

Referring to the class diagram designed in the Lab-4, we can identify the following classes:

- |                       |               |
|-----------------------|---------------|
| • SmartTableAssistant | • ServingTeam |
| • Customer            | • Cashier     |
| • Chef                | • Manager     |

The SmartTableAssistant asks the user for input every time when it is needed from the Customer class which is apparent. Also, minute details like manager advising the chef on his performance based on the food rating is also implemented which is apparent from the **improvise()** function (line 40, manager class).

The assistant generates a table ID which is used by the chef and the serving team to track the customer from whom the order has come. The serving team waits until the customer has finished eating food and then again asks if they need anything more.

## 7. Conclusions

Analysing the code and the requirements from the Lab-1, it can be concluded that the developed system can satisfy almost all the requirements.

## 8. Comments

**1. Limitations of Experiments**

None as noted.

**2. Limitations of Results**

None as noted.

**3. Learning happened**

In the current laboratory session, one could understand the practical aspect of a Software Development process. Unlike theoretical design, the practical software development offers much more challenge to the developer to implement the design.

**4. Recommendations**

None.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	