RESTAURANT MANAGEMENT SYSTEM

A PROJECT REPORT

Submited by

A.SHRIVALI (180101120034)

BANITA KUMARI SAHU (180101130001)

D.BHARGAVI (180101130004)

In partial fulfillment for the award of the degree

of



CENTURION UNIVERSITY OF

TECHNOLOGY & MANAGEMENT :: ODISHA

MONTH & YEAR

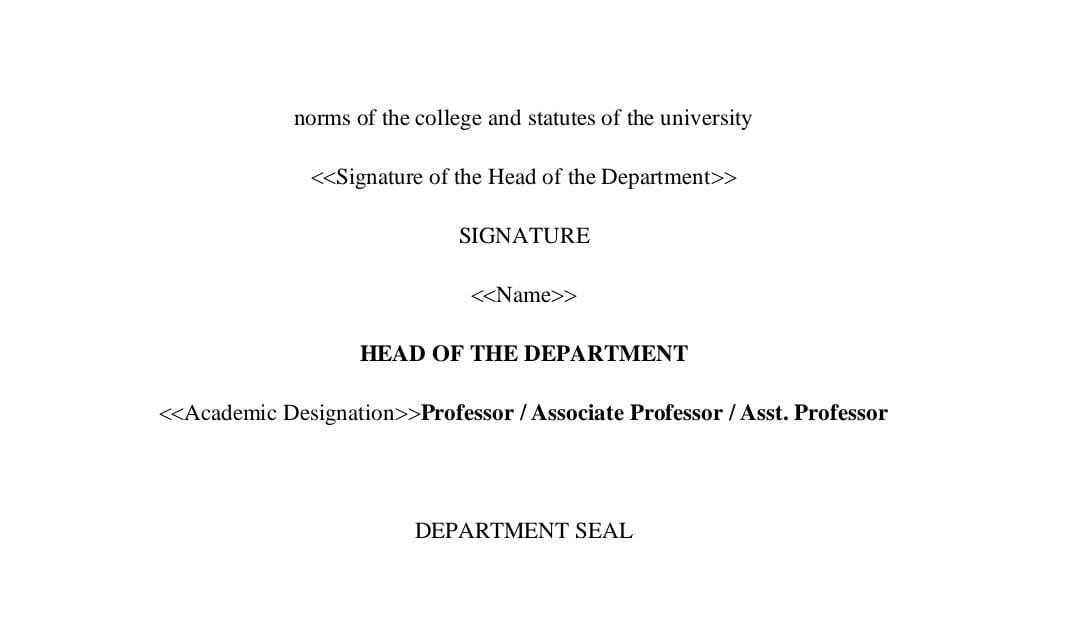
MAY 2021

Submitted by

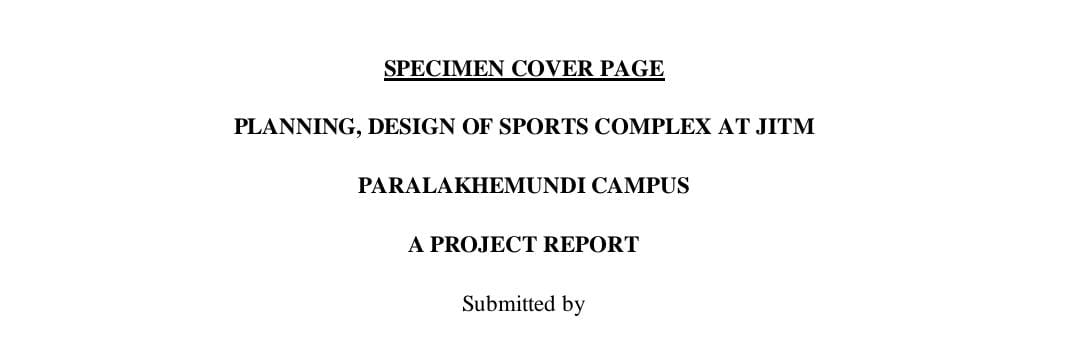
1.Banita kumari sahu(180101130001)

2.D.Bhargavi(180101130004)

3.A.Srivali(180101120034)



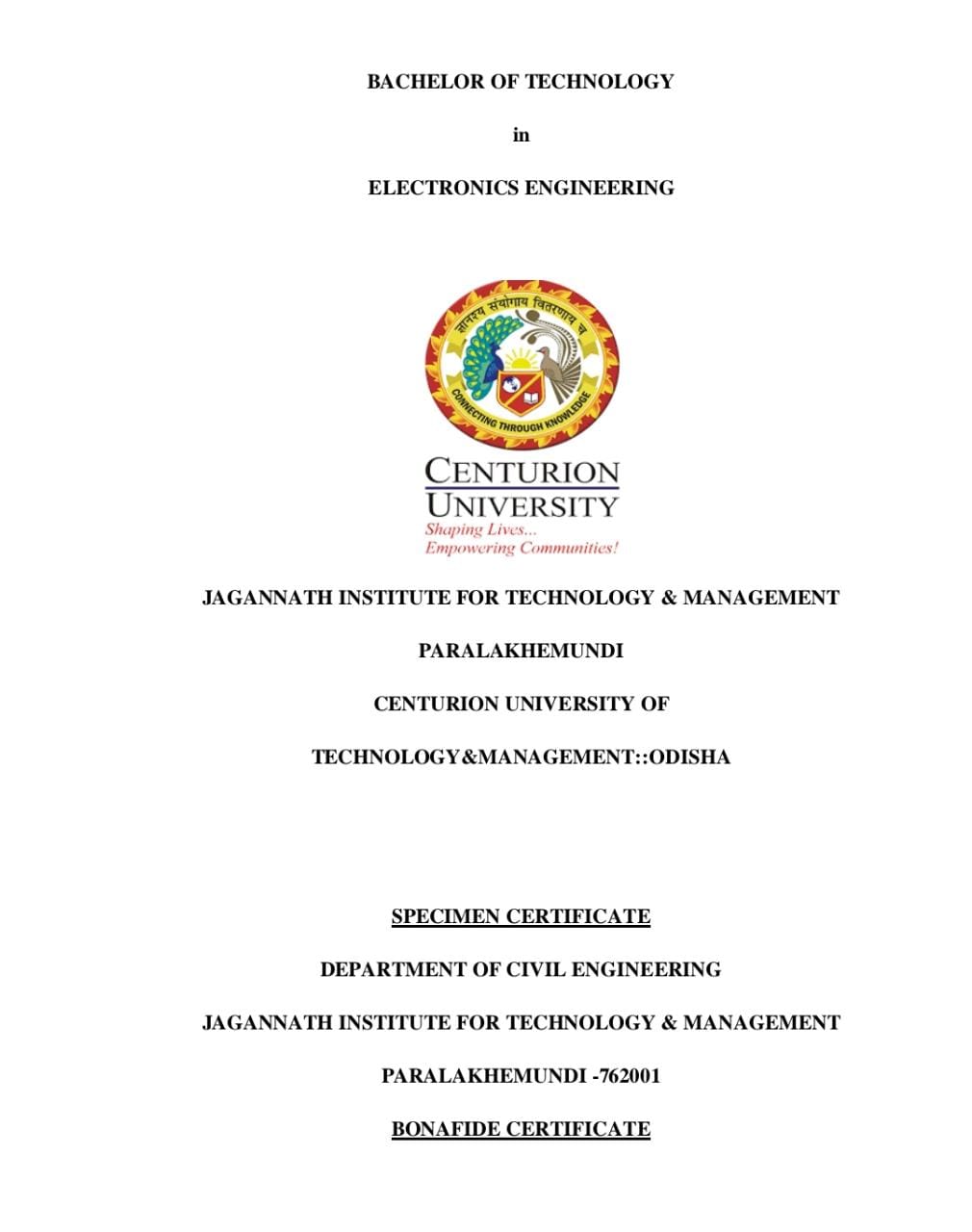
MAY 2021



BANITA KUMARI SAHU(180101130001)

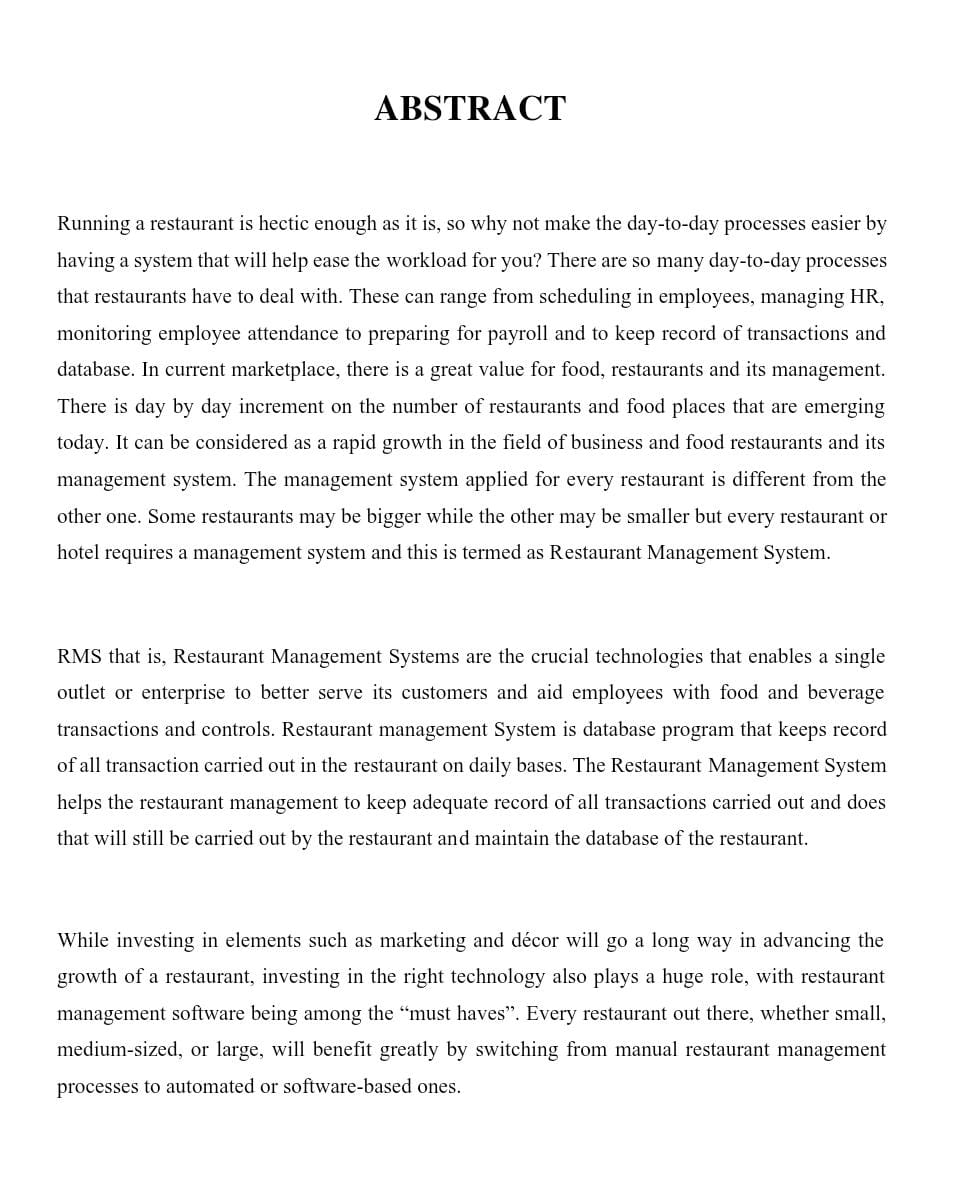
D.BHARGAVI(180101130004)

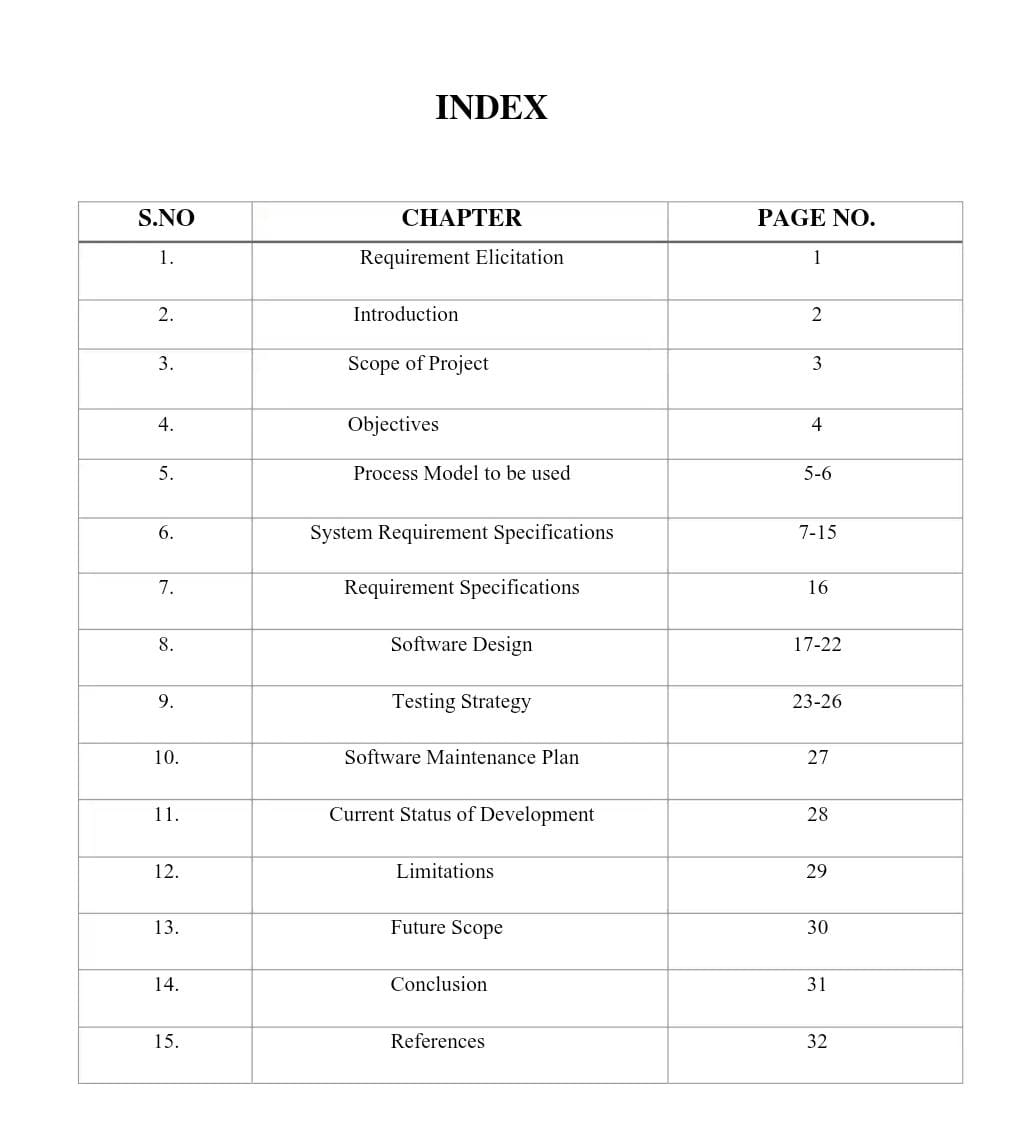
A.SHRIVALI(180101120034)

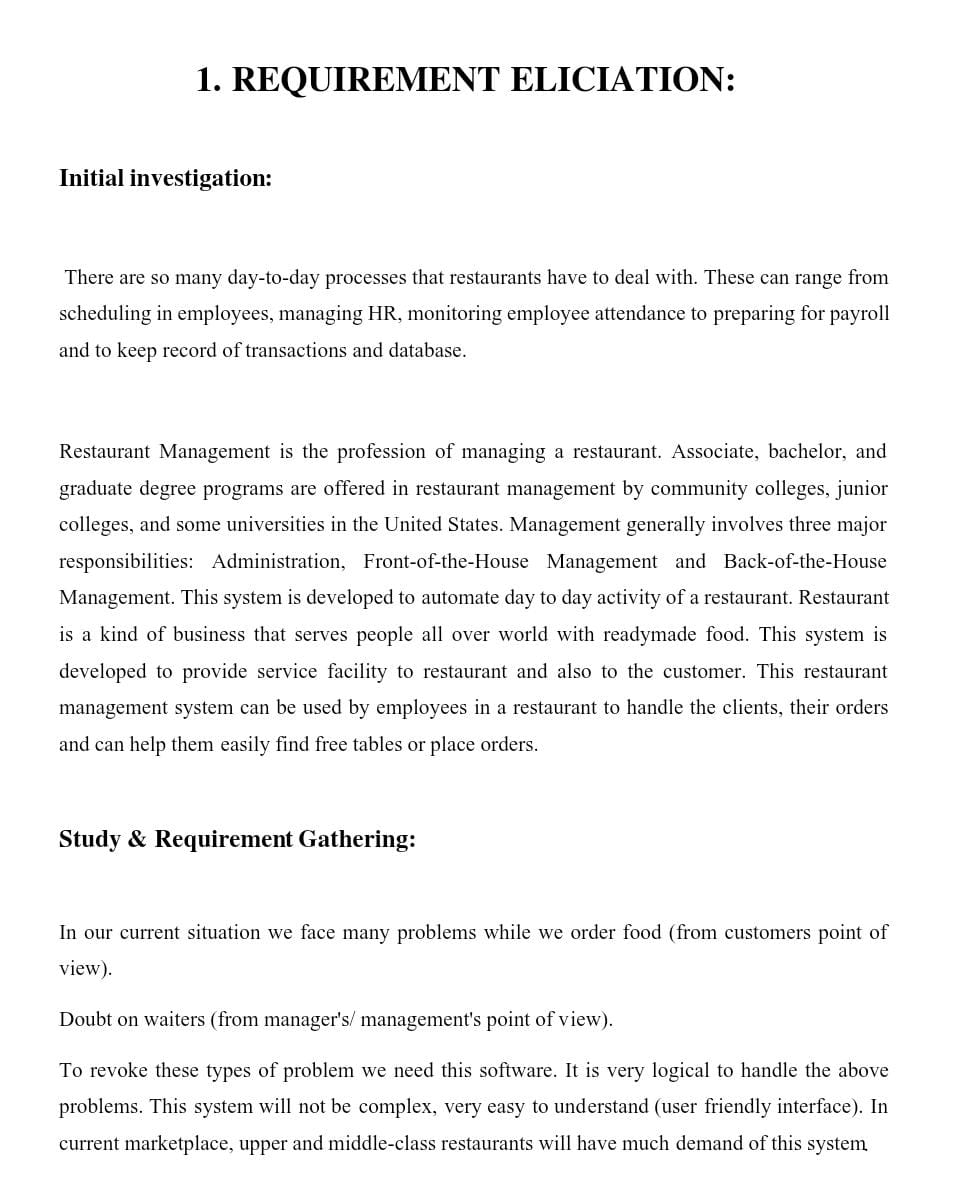
 ACKNOWLEDGEMENT

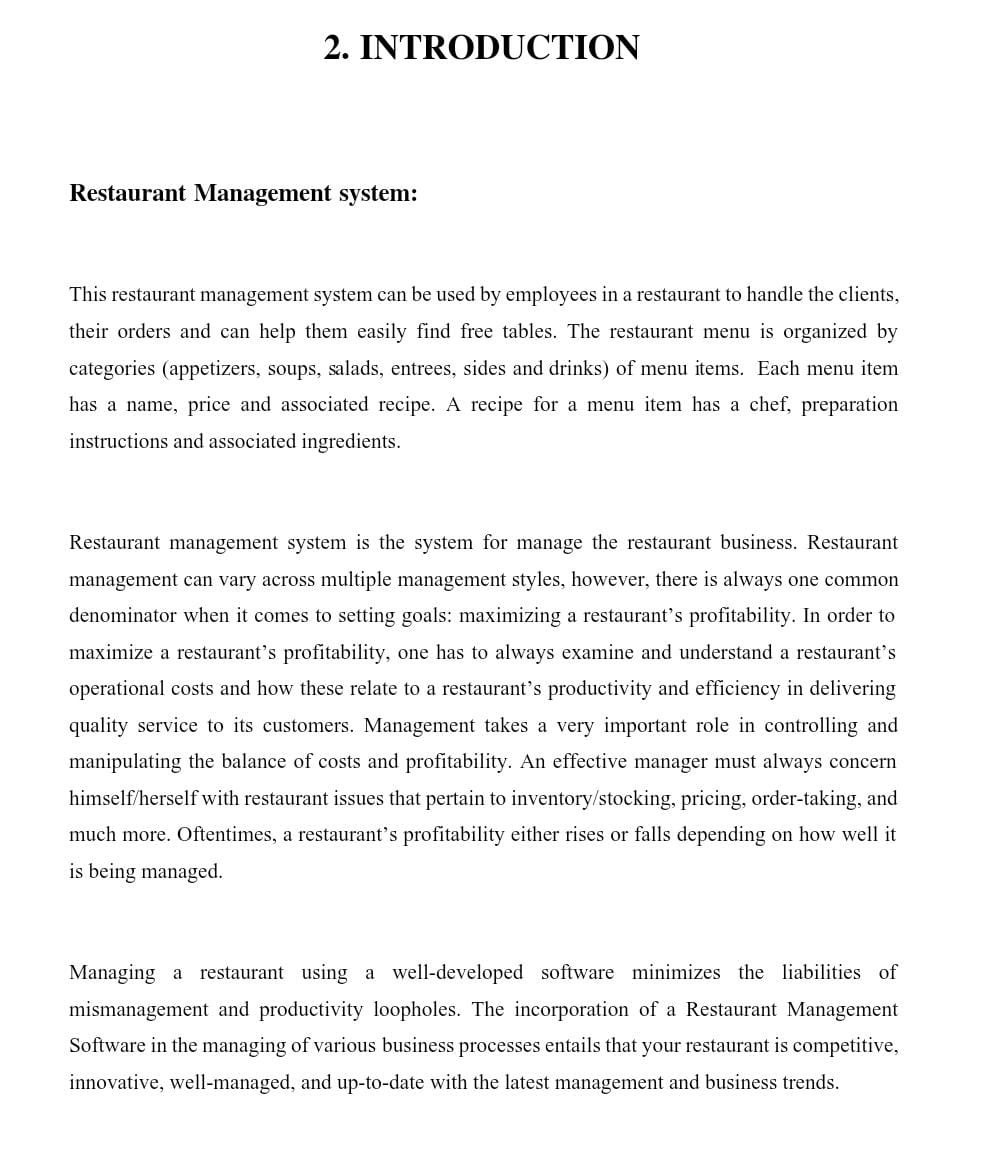
The completion of the Project work could have been possible with continued & dedicated efforts & guidance of MRS.SHIVANI NANDA mam of the college. We acknowledge our gratitude to her.

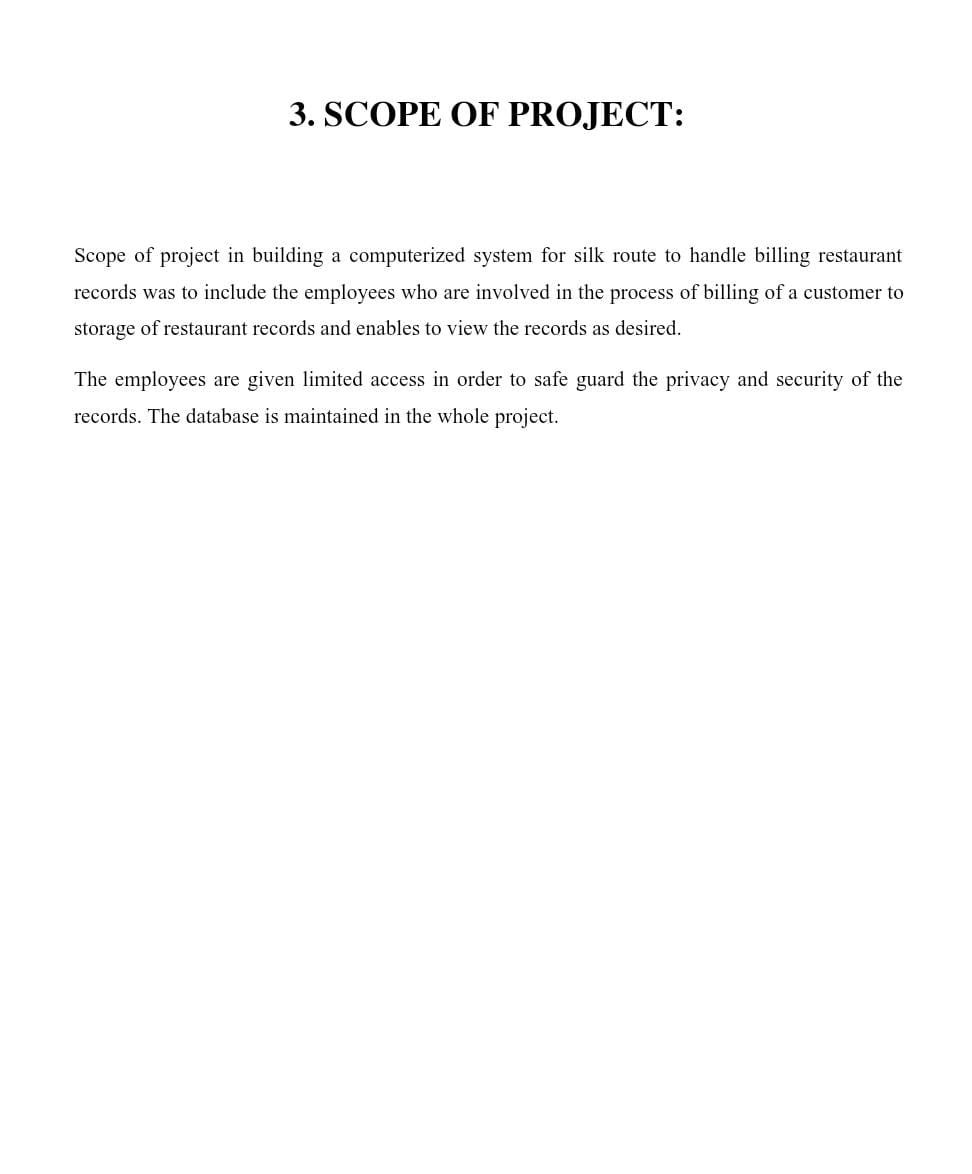
We wish to acknowledge our deep gratitude to MRS.SHIVANI NANDA MAM for her cooperation and guidance. It helped us in all the time of project work and writing of this Report. We could not have imagined having better advisor for our project work.

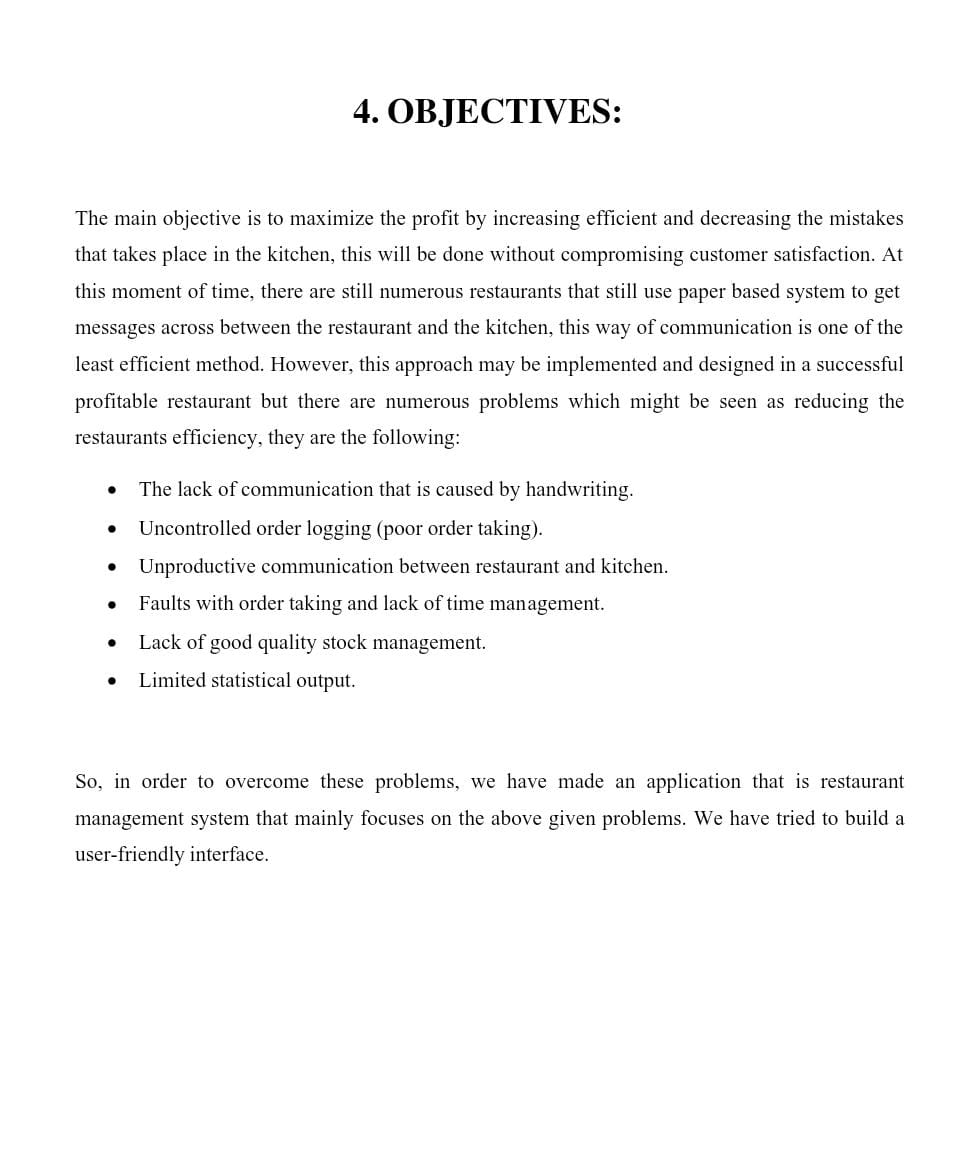




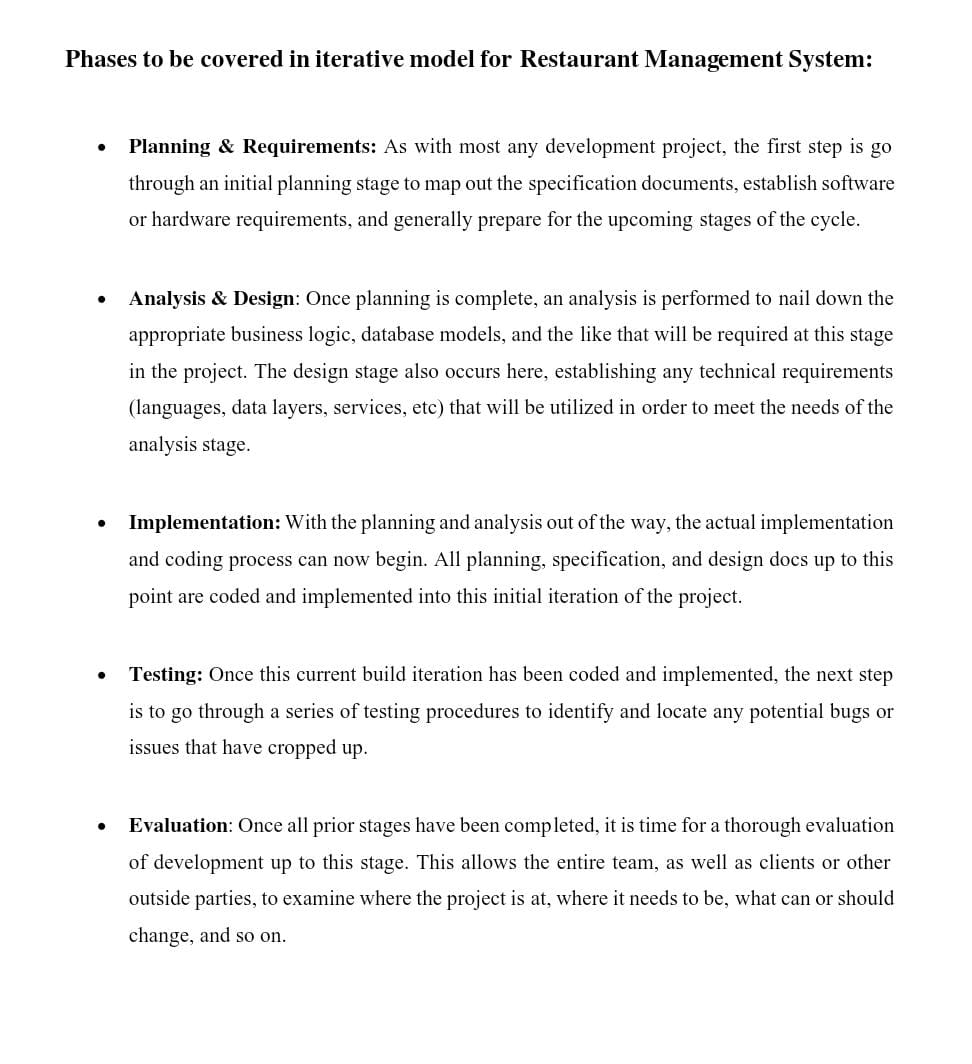


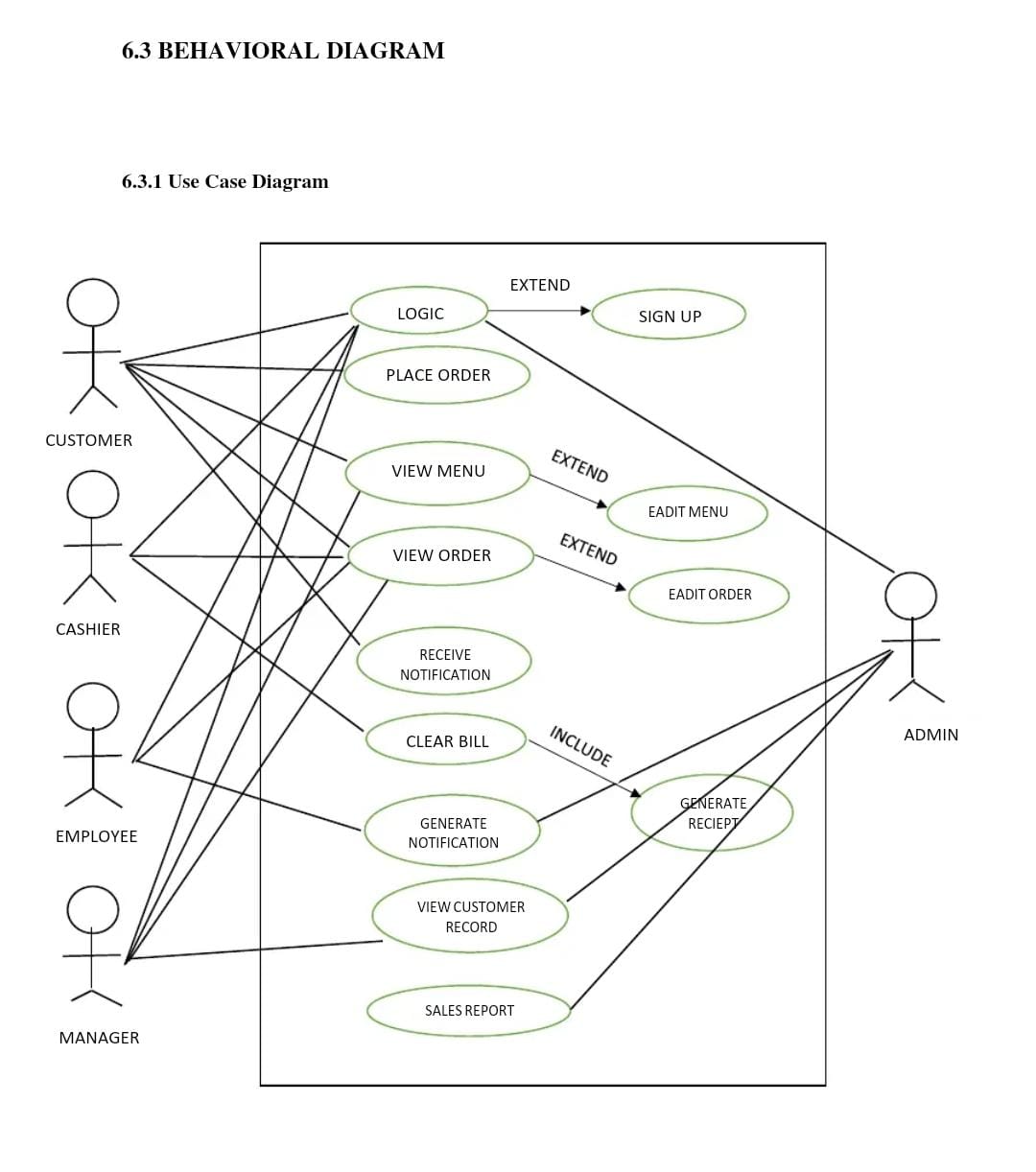


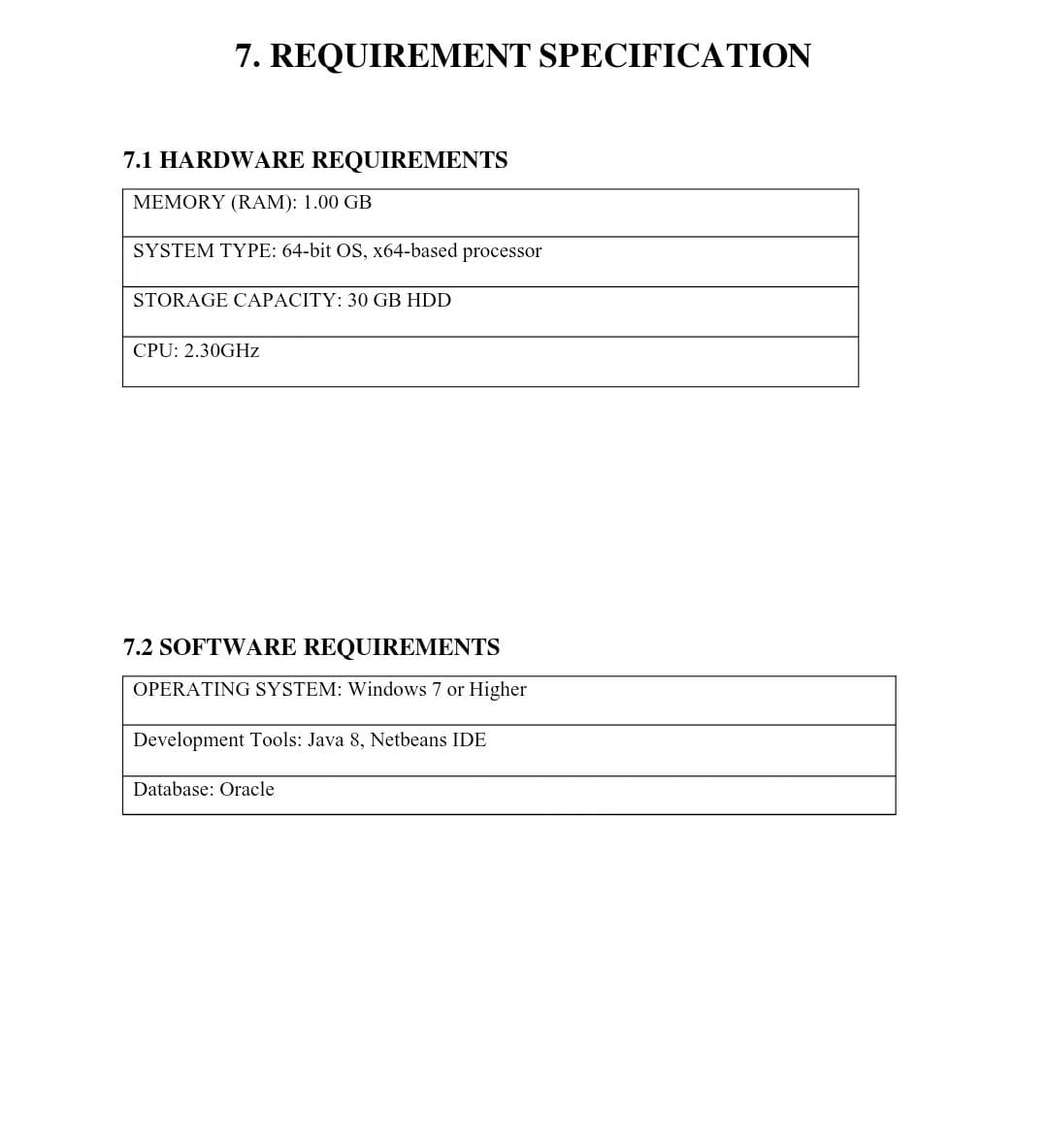


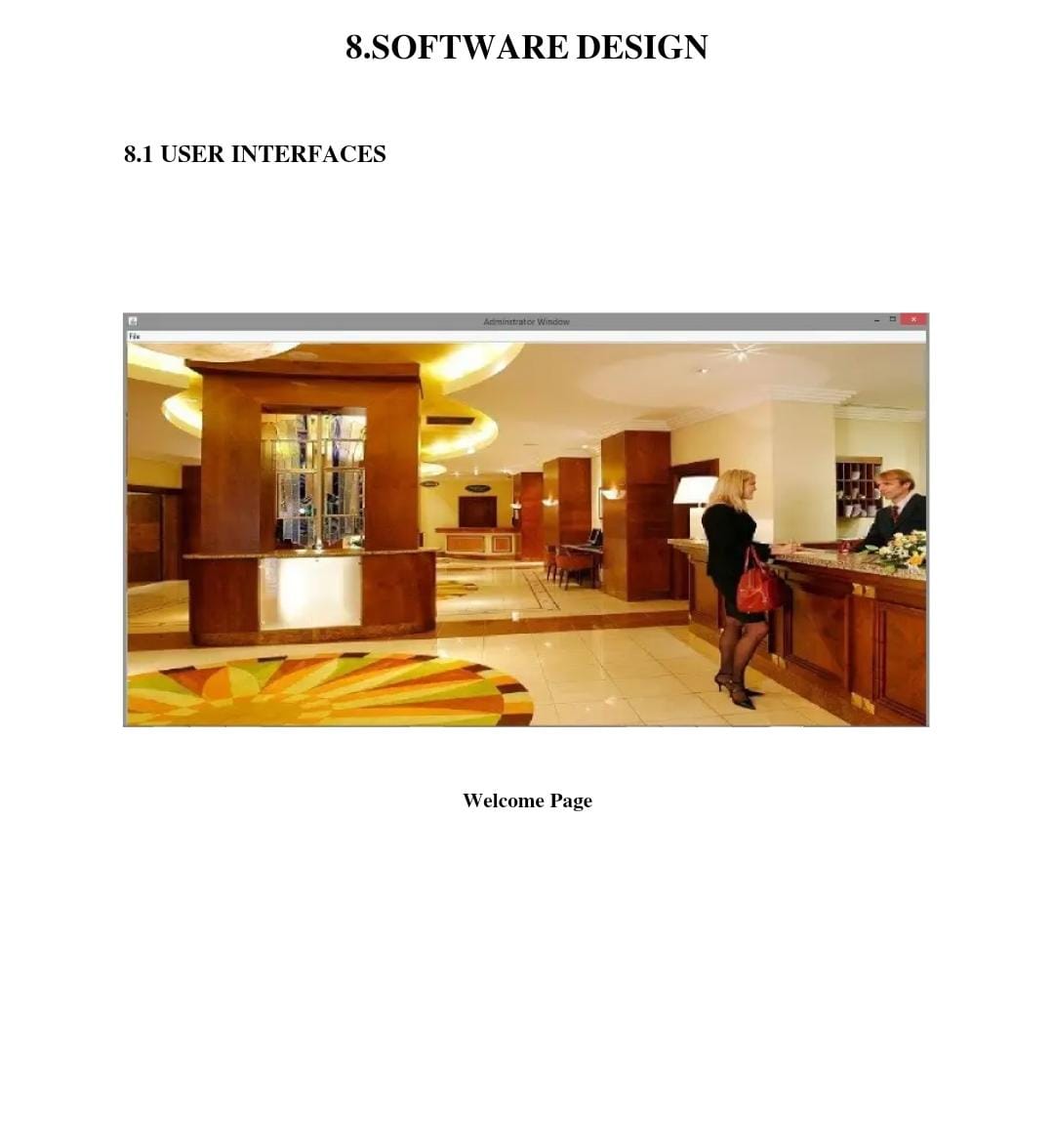


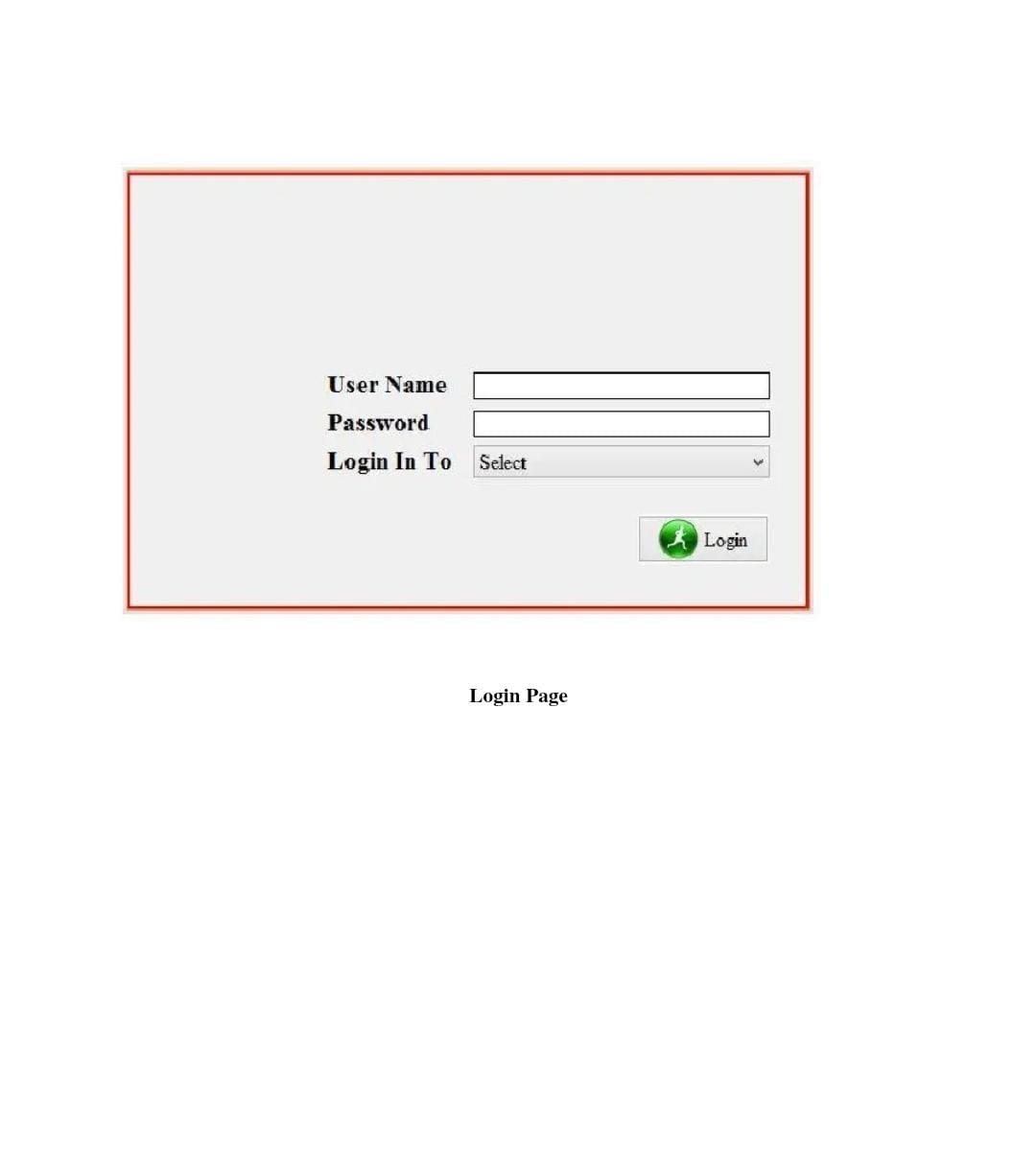






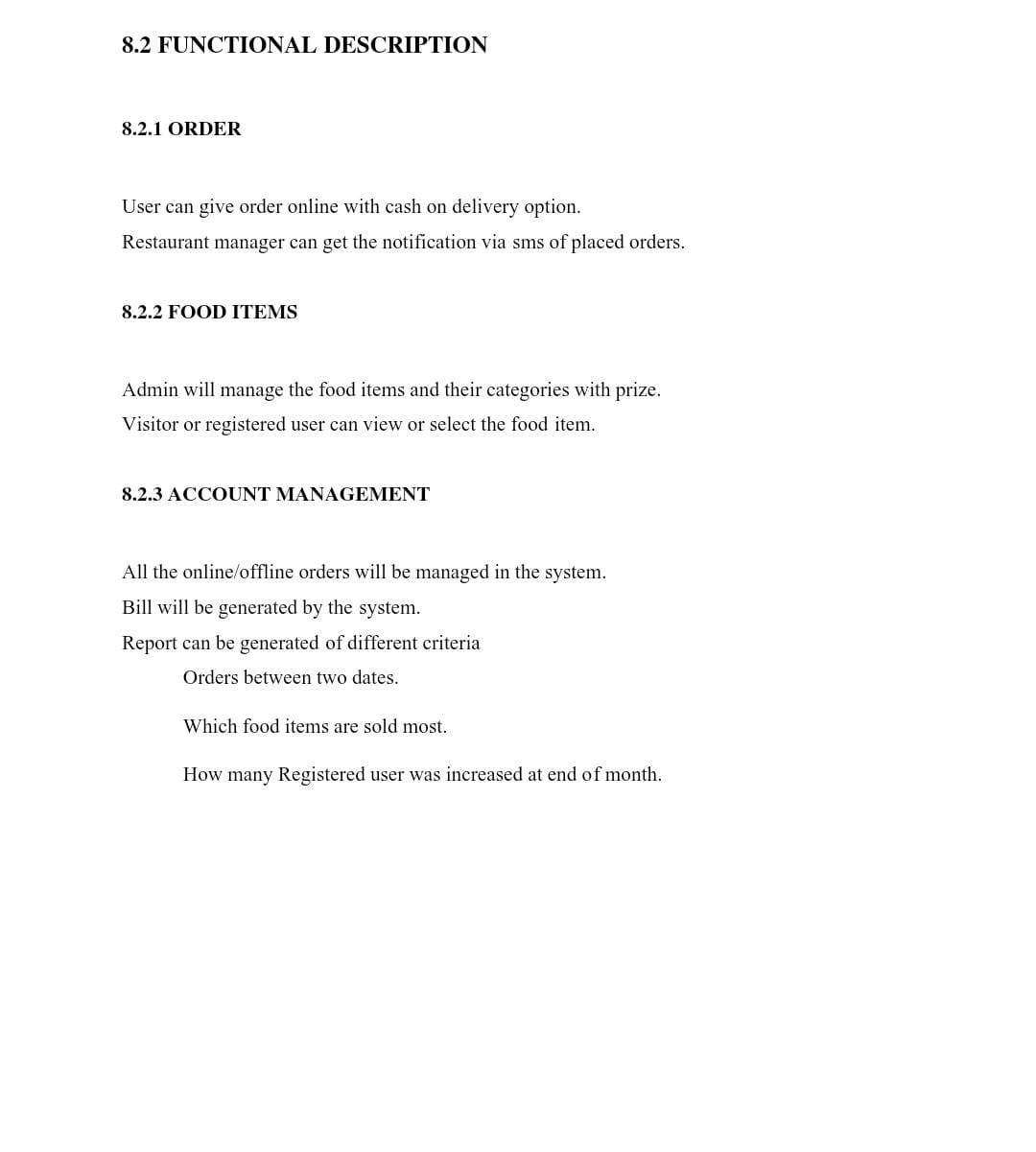


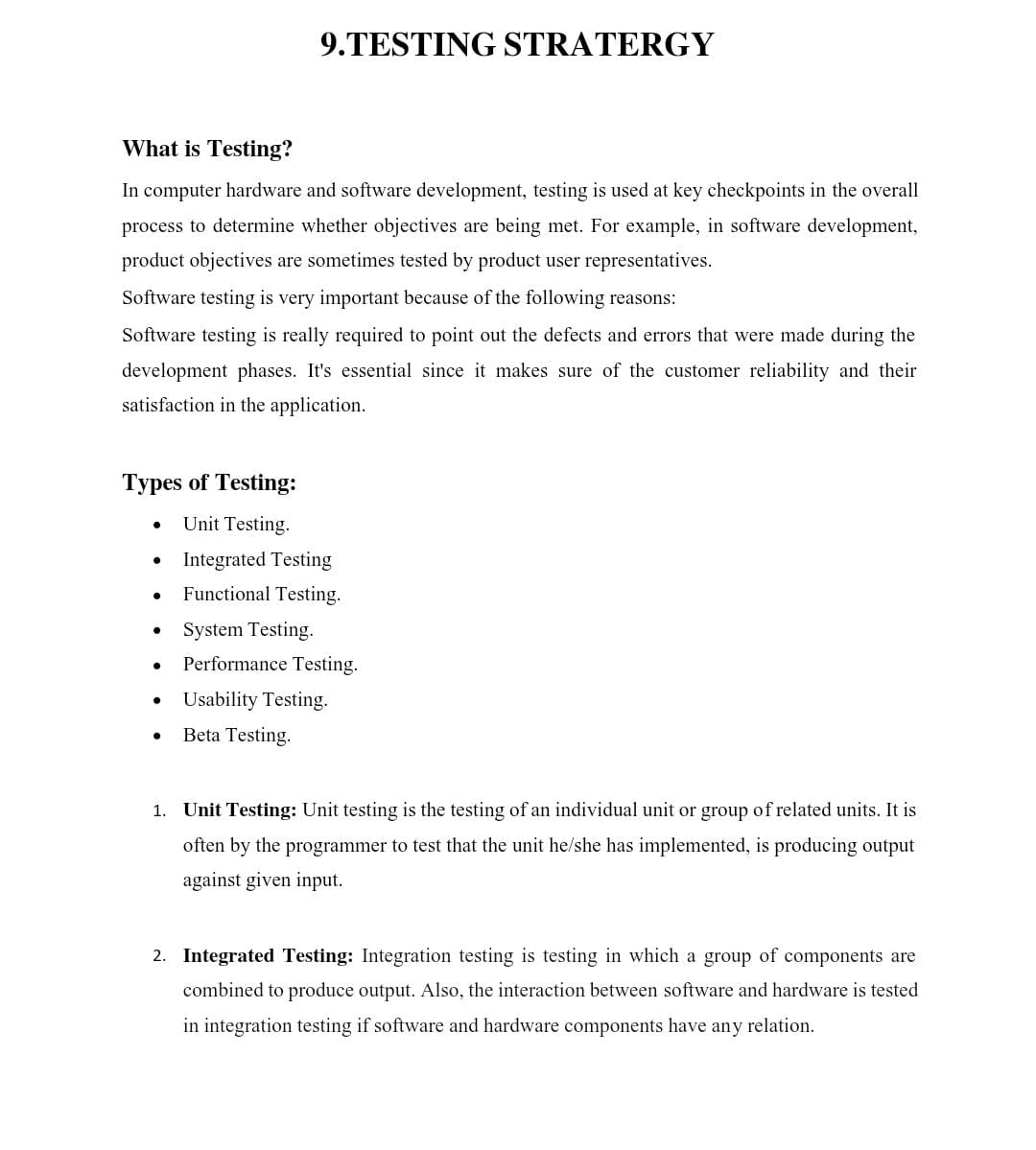


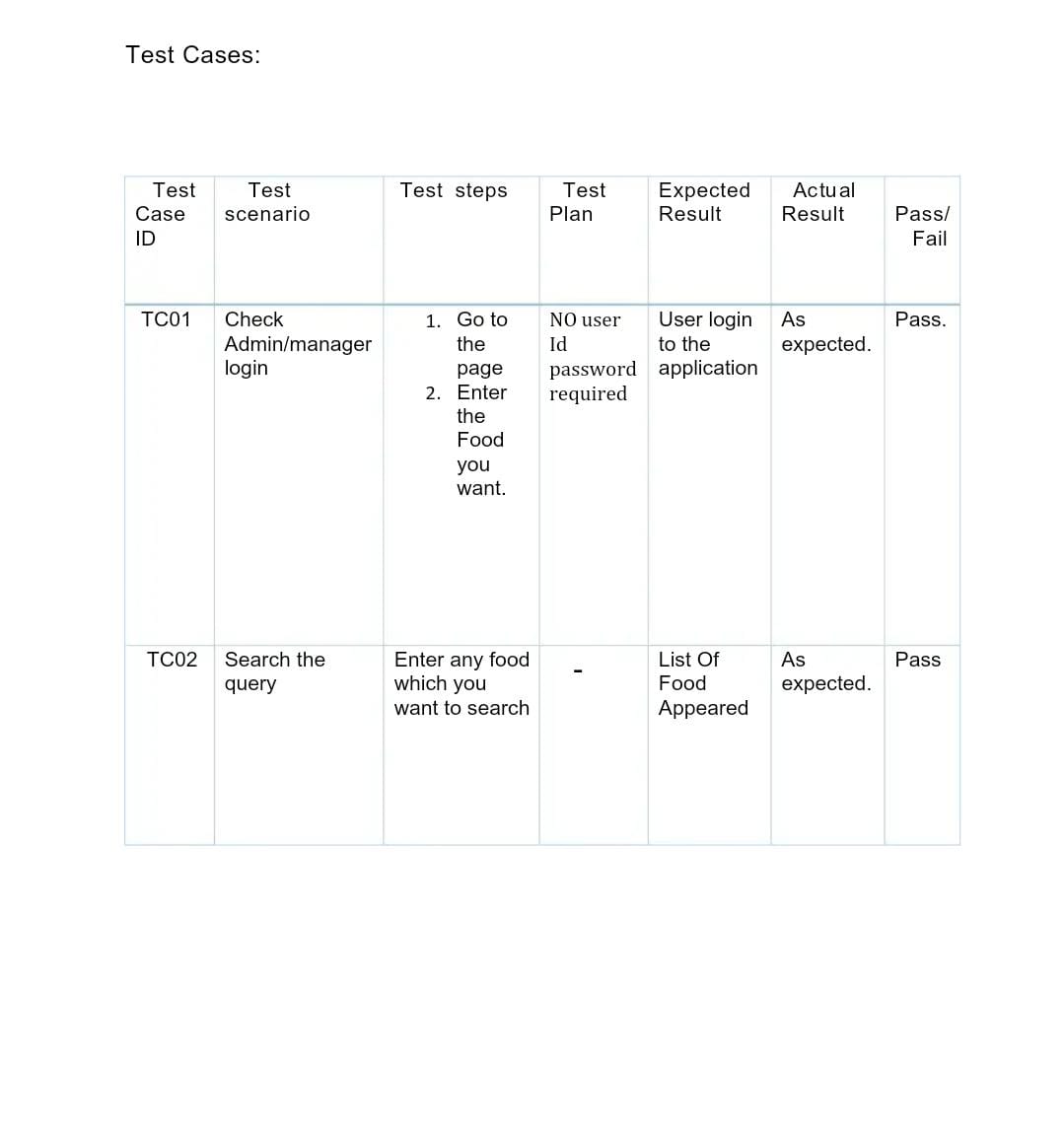


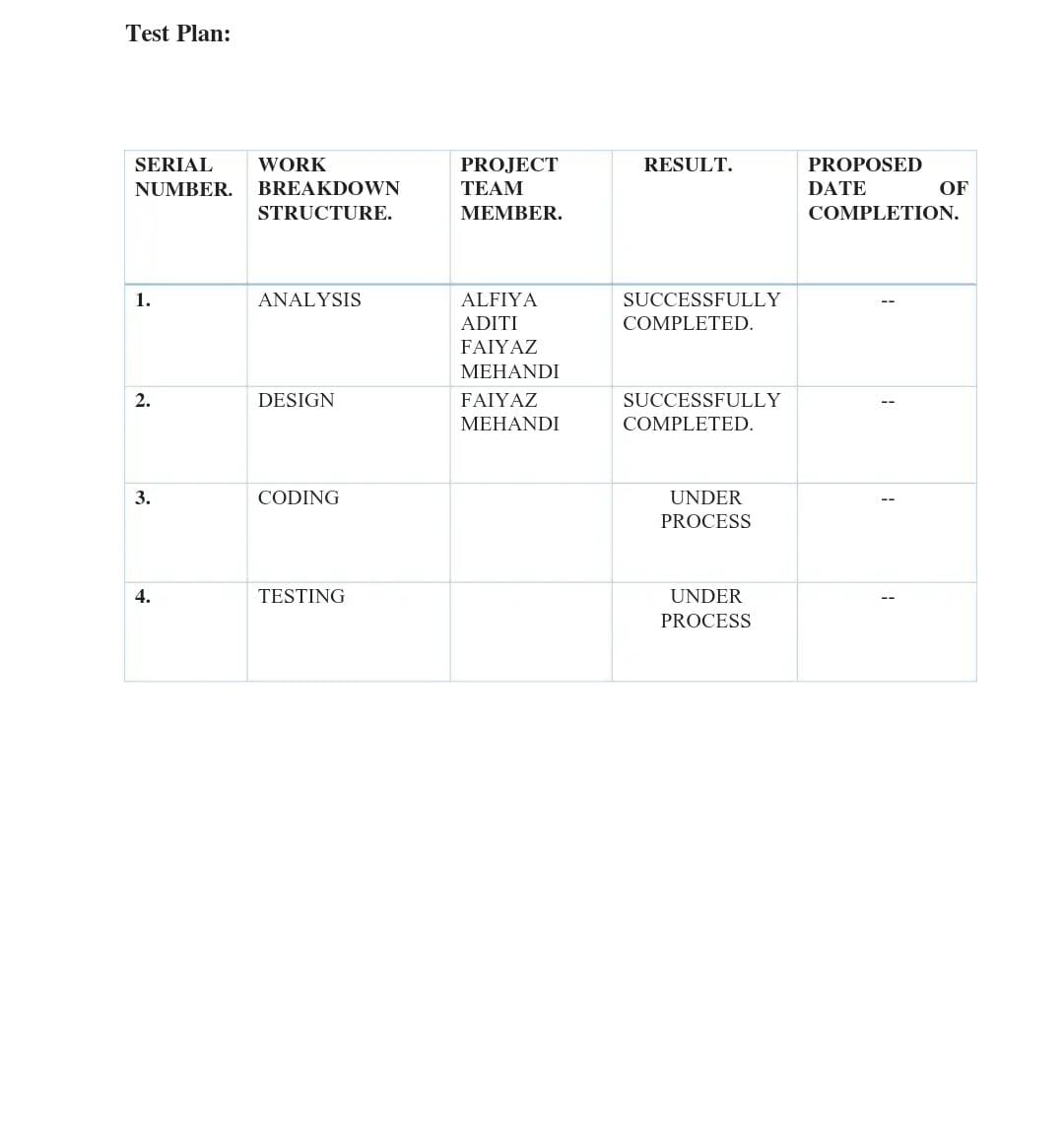


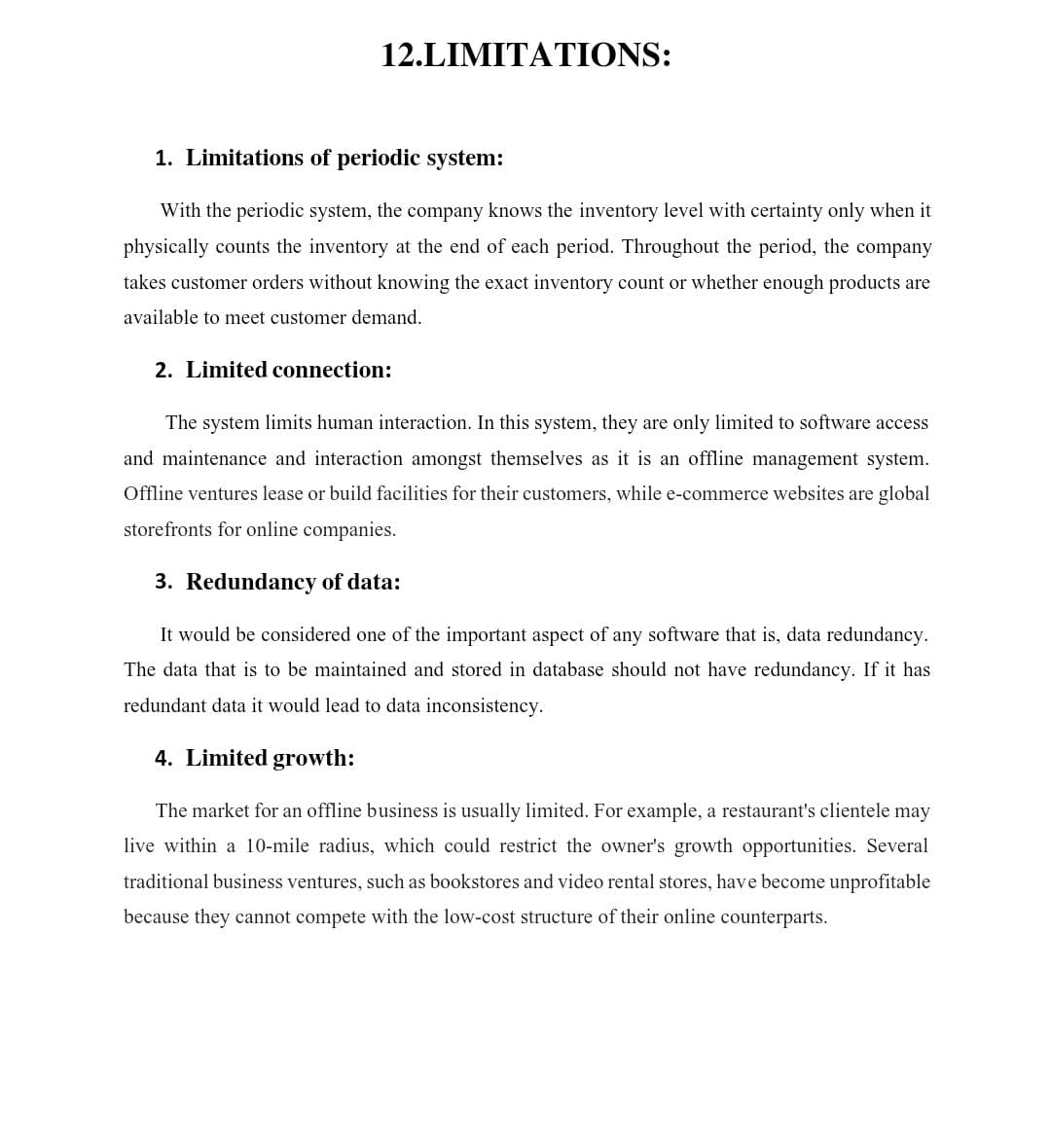


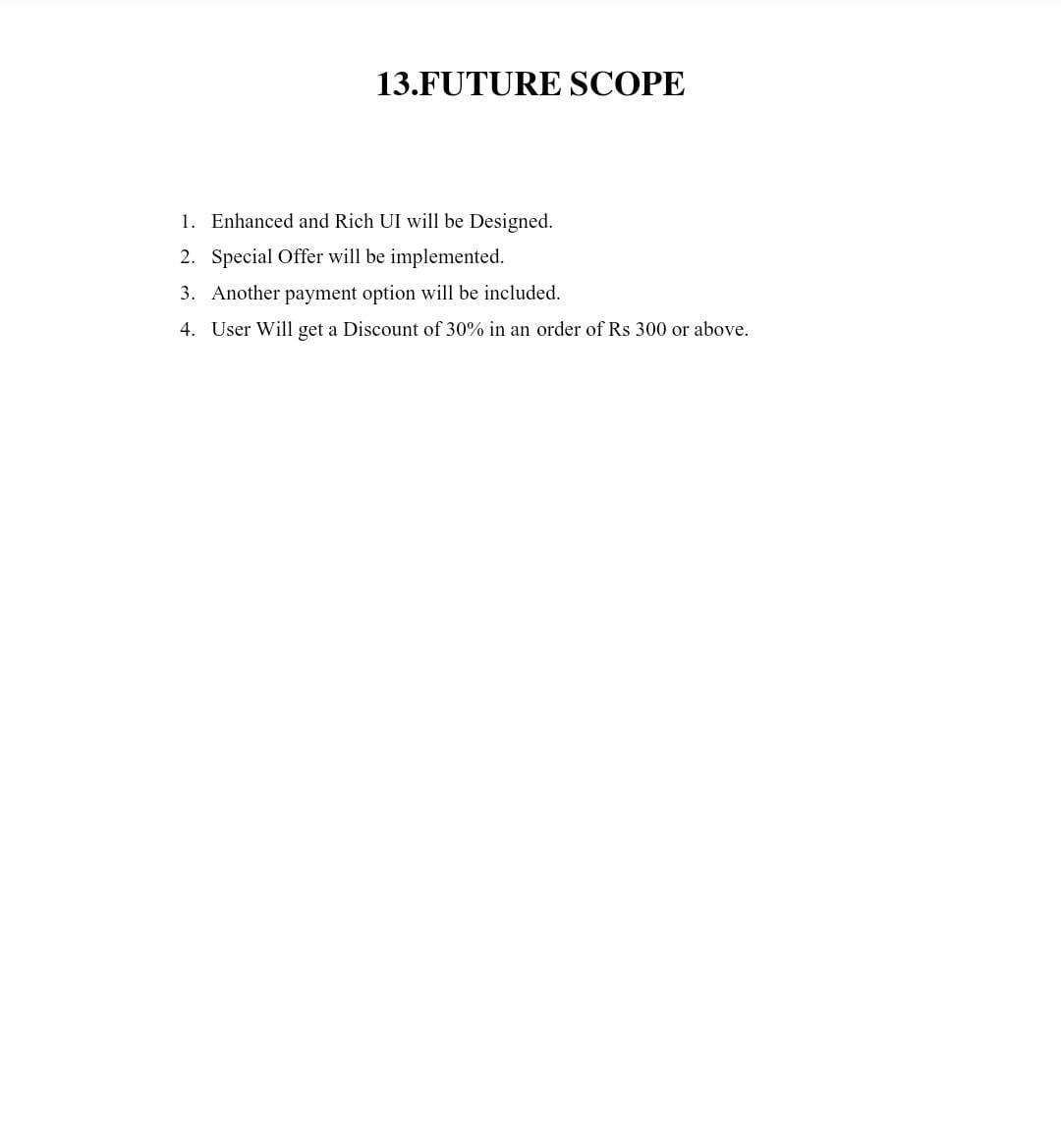












CODE:-

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct node

{

char foodname[50];

int quantity;

float price;

int data;

struct node \*prev;

struct node \*next;

};

struct node \*headc = NULL,\*newnode,\*tailc = NULL;

struct node \*heada = NULL, \*taila = NULL;

struct node \*head\_s;

void adminmenu()

{

printf("\n\t\t\t\t\t\t\t1. View total sales\n");

printf("\t\t\t\t\t\t\t2. Add new items in the order menu\n");

printf("\t\t\t\t\t\t\t3. Delete items from the order menu\n");

printf("\t\t\t\t\t\t\t4. Display order menu\n");

printf("\t\t\t\t\t\t\t5. Back To Main Menu \n\n");

printf("\t\t\t\t\t\t\t Enter Your Choice --->");

}

void customermenu()

{

printf("\n\t\t\t\t\t\t\t1. Place your order\n");

printf("\t\t\t\t\t\t\t2. View your ordered items\n");

printf("\t\t\t\t\t\t\t3. Delete an item from order\n");

printf("\t\t\t\t\t\t\t4. Display final bill\n");

printf("\t\t\t\t\t\t\t5. Back To Main Menu \n\n");

printf("\t\t\t\t\t\t\t Enter Your Choice --->");

}

struct node\* createadmin(struct node \*head,int data, char foodname[25], float price)

{

newnode = (struct node\*)malloc(sizeof(struct node));

newnode->data = data;

newnode->price = price;

newnode-> quantity = 0;

strcpy(newnode->foodname,foodname);

newnode->next = NULL;

newnode->prev = NULL;

struct node \*temp = head;

if(temp==NULL)

heada = taila = newnode;

else

{

while(temp->next!=NULL)

temp=temp->next;

temp->next=newnode;

newnode->prev = taila;

taila = newnode;

}

return heada;

}

struct node\* createcustomer(struct node \*head,int data,int quantity)

{

newnode = (struct node\*)malloc(sizeof(struct node));

struct node \*temp1 = heada;

int flag = 0;

while(temp1!=NULL)

{

if(temp1->data==data)

{

flag = 1;

break;

}

temp1 = temp1->next;

}

if(flag==1)

{

newnode->data = data;

newnode->price = quantity\*(temp1->price);

newnode-> quantity = quantity;

strcpy(newnode->foodname,temp1->foodname);

newnode->next = NULL;

newnode->prev = NULL;

struct node \*temp = head;

if(temp==NULL)

headc = tailc = newnode;

else

{

while(temp->next!=NULL)

temp=temp->next;

temp->next=newnode;

newnode->prev = tailc;

tailc = newnode;

}

}

else

{

printf("\n\t\t\t\t\t\t\tThis item is not present in the menu!\n");

}

return headc;

}

void displayList(struct node \*head)

{

struct node \*temp1 = head;

if(temp1==NULL)

{

printf("\n\t\t\t\t\t\t\t\tList is empty!!\n\n");

}

else

{

printf("\n");

while(temp1!=NULL)

{

if(temp1->quantity==0)

printf("\t\t\t\t\t\t\t%d\t%s\t%0.2f\n",temp1->data,temp1->foodname,temp1->price);

else

{

printf("\t\t\t\t\t\t\t%d\t%s\t%d\t%0.2f\n",temp1->data,temp1->foodname,temp1->quantity,temp1->price);

}

temp1 = temp1->next;

}

printf("\n");

}

}

struct node\* totalsales(int data,int quantity)

{

newnode = (struct node\*)malloc(sizeof(struct node));

int flag = 0;

struct node \*temp1 = heada;

while(temp1->data!=data)

{

temp1 = temp1->next;

}

newnode->data = data;

newnode->price = quantity\*(temp1->price);

newnode-> quantity = quantity;

strcpy(newnode->foodname,temp1->foodname);

newnode->next = NULL;

newnode->prev = NULL;

struct node \*temp = head\_s;

if(temp==NULL)

head\_s = newnode;

else

{

while(temp->next!=NULL)

{

if(temp->data==data)

{

flag = 1;

break;

}

temp=temp->next;

}

if(flag==1)

{

temp->quantity += newnode-> quantity;

temp->price += newnode->price;

}

else

{

temp->next=newnode;

}

}

return head\_s;

}

void calculatetotsales()

{

struct node \*temp = headc;

while(temp!=NULL)

{

head\_s = totalsales(temp->data, temp->quantity);

temp=temp->next;

}

}

struct node\* delete(int data,struct node \*head, struct node\* tail)

{

if(head==NULL)

{

printf("\n\t\t\t\t\t\t\tList is empty\n");

}

else

{

struct node\* temp;

if(data==head->data)

{

temp = head;

head = head->next;

if (head != NULL)

head->prev = NULL;

free(temp);

}

else if(data==tail->data)

{

temp = tail;

tail = tail->prev;

tail->next = NULL;

free(temp);

}

else

{

temp = head;

while(data!=temp->data)

{

temp = temp->next;

}

(temp->prev)->next = temp->next;

(temp->next)->prev = temp->prev;

free(temp);

}

}

return head;

}

int deleteadmin()

{

printf("\n\t\t\t\t\tEnter serial no. of the food item which is to be deleted: ");

int num;

scanf("%d",&num);

struct node\* temp=heada;

while(temp!=NULL)

{

if (temp->data == num)

{

heada = delete(num, heada, taila);

return 1;

}

temp=temp->next;

}

return 0;

}

int deletecustomer()

{

printf("\n\t\t\t\t\tEnter serial no. of the food item which is to be deleted: ");

int num;

scanf("%d",&num);

struct node\* temp=headc;

while(temp!=NULL)

{

if (temp->data == num)

{

headc = delete(num, headc, tailc);

return 1;

}

temp=temp->next;

}

return 0;

}

void displaybill()

{

displayList(headc);

struct node \*temp = headc;

float total\_price = 0;

while (temp!=NULL)

{

total\_price +=temp->price;

temp = temp->next;

}

printf("\t\t\t\t\t\t\tTotal price: %0.02f\n",total\_price);

}

struct node\* deleteList(struct node\* head)

{

if(head==NULL)

{

return NULL;

}

else

{

struct node\* temp = head;

while(temp->next!=0)

{

temp = temp->next;

free(temp->prev);

}

free(temp);

head = NULL;

}

return head;

}

void admin()

{

printf("\n\t\t\t\t\t ----------------------------------------------\n");

printf("\t\t\t\t\t\t\t ADMIN SECTION\n");

printf("\t\t\t\t\t ----------------------------------------------\n");

while(1)

{

adminmenu();

int opt;

scanf("%d",&opt);

if(opt==5)

break;

switch (opt)

{

case 1:

displayList(head\_s);

break;

case 2:

printf("\n\t\t\t\t\t\t\tEnter serial no. of the food item: ");

int num,flag = 0;

char name[50];

float price;

scanf("%d",&num);

struct node \*temp = heada;

while(temp!=NULL)

{

if(temp->data==num)

{

printf("\n\t\t\t\t\t\tFood item with given serial number already exists!!\n\n");

flag = 1;

break;

}

temp = temp->next;

}

if(flag==1)

break;

printf("\t\t\t\t\t\t\tEnter food item name: ");

scanf("%s",name);

printf("\t\t\t\t\t\t\tEnter price: ");

scanf("%f",&price);

heada = createadmin(heada, num, name, price);

printf("\n\t\t\t\t\t\t\tNew food item added to the list!!\n\n");

break;

case 3:

if(deleteadmin())

{

printf("\n\t\t\t\t\t\t### Updated list of food items menu ###\n");

displayList(heada);

}

else

printf("\n\t\t\t\t\t\tFood item with given serial number doesn't exist!\n\n");

break;

case 4:

printf("\n\t\t\t\t\t\t\t ### Order menu ###\n");

displayList(heada);

break;

default:

printf("\n\t\t\t\t\t\tWrong Input !! PLease choose valid option\n");

break;

}

}

}

void customer()

{

int flag=0,j=1;

char ch;

printf("\n\t\t\t\t\t ----------------------------------------------\n");

printf("\t\t\t\t\t\t\t CUSTOMER SECTION\n");

printf("\t\t\t\t\t ----------------------------------------------\n");

while(1)

{

customermenu();

int opt;

scanf("%d",&opt);

if(opt==5)

break;

switch (opt)

{

case 1:

displayList(heada);

printf("\n\t\t\t\t\t\tEnter number corresponding to the item you want to order: ");

int n;

scanf("%d",&n);

printf("\t\t\t\t\t\tEnter quantity: ");

int quantity;

scanf("%d",&quantity);

headc = createcustomer(headc, n, quantity);

break;

case 2:

printf("\n\t\t\t\t\t\t\t ### List of ordered items ###\n");

displayList(headc);

break;

case 3:

if(deletecustomer())

{

printf("\n\t\t\t\t\t\t### Updated list of your ordered food items ###\n");

displayList(headc);

}

else

printf("\n\t\t\t\t\t\tFood item with given serial number doesn't exist!!\n");

break;

case 4:

calculatetotsales();

printf("\n\t\t\t\t\t\t\t ### Final Bill ###\n");

displaybill();

headc = deleteList(headc);

printf("\n\t\t\t\t\t\tPress any key to return to main menu:\n\t\t\t\t\t\t");

fflush(stdin);

ch=fgetc(stdin);

flag=1;

break;

default:

printf("\n\t\t\t\t\t\tWrong Input !! PLease choose valid option\n");

break;

}

if(flag==1)

break;

}

}

void mainnenu()

{

printf("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" WELCOME TO RESTAURANT MANAGEMENT SYSTEM\n");

printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n");

printf("\t\t\t\t\t\t\t1. ADMIN SECTION--> \n");

printf("\t\t\t\t\t\t\t2. CUSTOMER SECTION--> \n");

printf("\t\t\t\t\t\t\t3. Exit--> \n\n");

printf("\t\t\t\t\t\t\tEnter Your Choice --->");

}

int main()

{

heada = createadmin(heada,1,"Hot and Sour Soup",100);

heada = createadmin(heada,2,"Manchow Soup",200);

heada = createadmin(heada,3,"Manchurian Noodles",150);

heada = createadmin(heada,4,"Fried Rice",180);

heada = createadmin(heada,5,"Hakka Noodles",80);

while(1)

{

mainnenu();

int choice;

scanf("%d",&choice);

if(choice==3)

{

printf("\n\n\t\t\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*Thank you!!\*\*\*\*\*\*\*\*\*\*\n");

break;

}

switch (choice)

{

case 1:

admin();

break;

case 2:

customer();

break;

case 3:

break;

default:

printf("\n\t\t\t\t\t\tWrong Input !! PLease choose valid option\n");

break;

}

}

}

OUTPUTS:-

