Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report on

"Food Ordering System using Data Structure"

Submitted by:

Saugat Bhattarai (04)

Abhay Sharma (17)

Pratik K. Khanal(10)

Submitted to: Prakash Poudel

Department of Computer Science and Engineering

Submission date: 10th of September 2021

Abstract:

Through the study of COMP 202 (Data Structures and Algorithms) and gaining crucial knowledge about different kinds of data structures like stacks , queues , linked lists BSTs and so on, we have undoubtedly increased our horizons in the specter of programming. Having said that, writing a program using said data structures was indeed a daunting task with various hurdles that we encountered along the way.

Nevertheless , we managed to use a crucial data structure(linked lists) to use as the building block of our program. The project most certainly helped us to improve our understanding of data structures while also expanding our programming temperament.

The motivation behind choosing to program a food ordering system isn't a specific incident but inconveniences faced by students like us while trying to order food around the university. This gave us the idea to implement this program using data structures.

We have decided to use the C programming language to undertake this project due to its rich inbuilt library which provides a large number of built-in functions. Also, C programming was the language used in class and therefore we found it more comfortable to use C instead of python which was our initial option.

The principle outcome we expected from this project were a smooth functioning, easy to use program using linked lists and a better understanding of the concept of data structures.

To conclude, though there could be improvements made in the project, considering all the time constraints and complexities, we are satisfied with how the project turned out to be.

TABLE OF CONTENTS

Chapter 1: Introduction

The aim of developing FOOD ORDERING SYSTEM project is to replace the traditional way of taking orders with computerized system. Another important reason for developing this project is to prepare order summary reports quickly and in correct format at any point of time when required. FOOD ORDERING SYSTEM has a very lot of scope. This project is easy, fast and accurate. It requires less disk space.

- **User Friendly:** FOOD ORDERING SYSTEM is a very user friendly project because the Food Ordering user interface of the project is very simple.

- **Order reports** of the system can be easily generated. User can generate the bill after ordering. In this way they can get delivery status of customers and get information about what is being ordered.

- **Very less paper work**: FOOD ORDERING SYSTEM requires less paper work. In this project all record is fetched directly into the computer and reports can be generated through just a click. In this way it saves time. As data is directly entered into computer so there is no need to do any paper work.4.

- **Computer operator control**: FOOD ORDERING SYSTEM is operated by the staff members and one admin so there is no chance of clerical mistakes. Data feeding andretrieving in this system is very easy. So the work can be done on time.

Background

This project is a basic implementation of data structures. The service offered by food ordering systems these days and the ease they offer was the primary idea behind the program.

Objectives

The main objectives of the project can be summed up as follows:

- **A smooth functioning program using data structures**: The main achievement we had hoped for after the completion of the project was a smoothly running program, with minimal bugs and a considerable time complexity.
- **A gain in programming fluency**: an improved programming knowledge was another objective we had hoped to attain. Having the opportunity to work as a team to complete this project also helped immensely to improve teamwork.
- **A clear concept on the usage and details of data structures**: having to use data structures to do the project we had hoped to more properly and clearly understand the concept as a primary objective to attain

Motivation

When given the task to choose the topic for our project, we were in a state of confusion to decide on an appropriate topic. As while being assigned the task we had only learned about stacks, queues and a bit about Linked lists. Therefore, we chose to use linked lists as the basic data structure for our program.

Theory

A linked list is a sequence of data structures, which are connected together via links.

Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.

- **Link** − Each link of a linked list can store a data called an element.
- **Next** − Each link of a linked list contains a link to the next link called Next.
- **LinkedList** − A Linked List contains the connection link to the first link called First.
-

Linked List Representation

Linked list can be visualized as a chain of nodes, where every node points to the next node.

As per the above illustration, following are the important points to be considered.

- Linked List contains a link element called first.
- Each link carries a data field(s) and a link field called next.
- Each link is linked with its next link using its next link.
- Last link carries a link as null to mark the end of the list.

Types of Linked List

Following are the various types of linked list.

- **Simple Linked List** − Item navigation is forward only.
- **Doubly Linked List** − Items can be navigated forward and backward.
- **Circular Linked List** − Last item contains link of the first element as next and the first element has a link to the last element as previous.

Basic Operations

Following are the basic operations supported by a list.

- **Insertion** − Adds an element at the beginning of the list.
- **Deletion** − Deletes an element at the beginning of the list.
- **Display** − Displays the complete list.
- **Search** − Searches an element using the given key.
- **Delete** − Deletes an element using the given key.

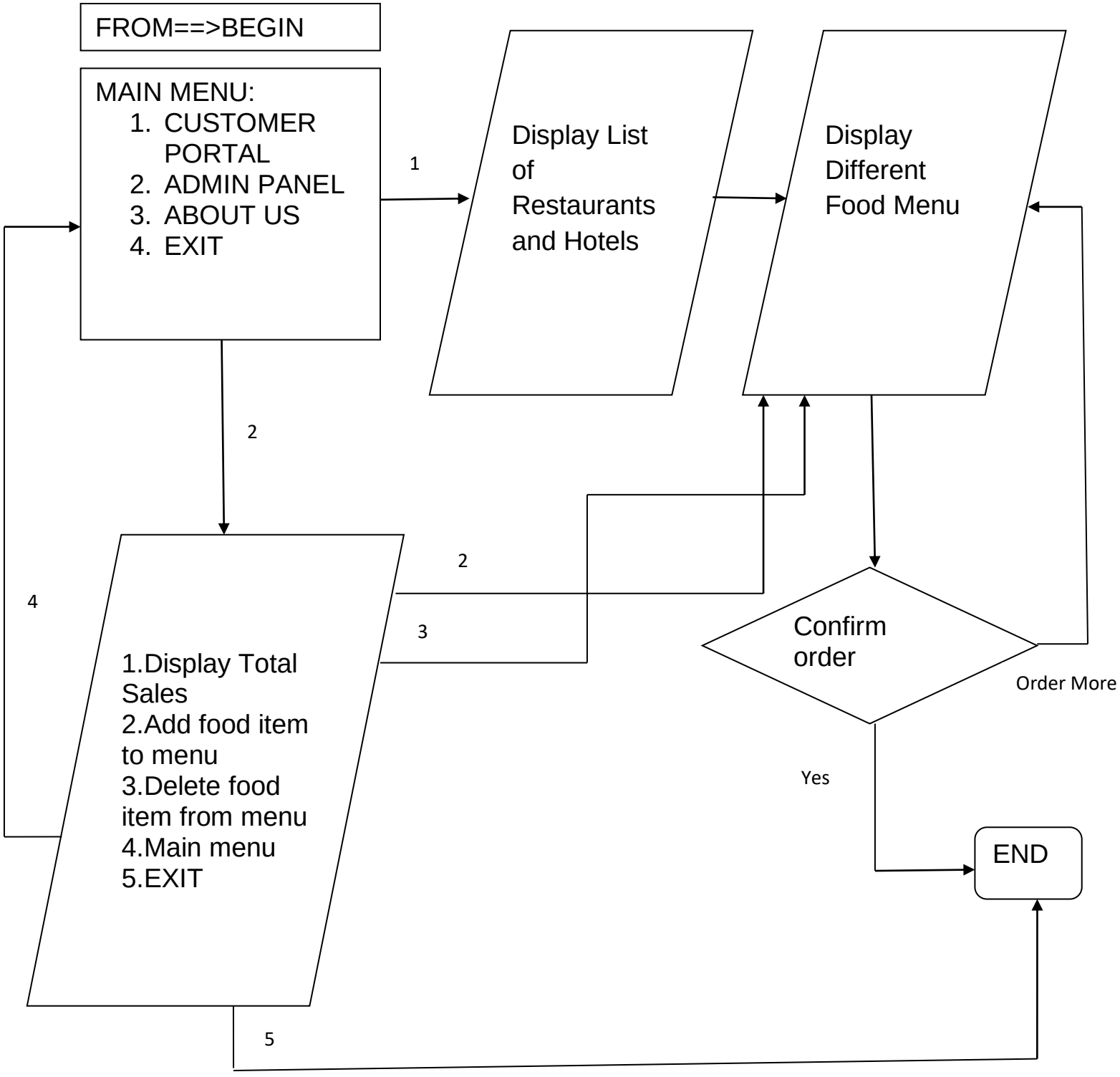
Chapter 2: Design and Implementation

The design of our program is simple yet effective. Our main goal was to keep the program's design basic in order for everyone to understand it.

Discussing about the design, the first interface we come across when the program is run is the main menu. This section contains four sub-divisions:

- Customer section: on choosing this section the user is met with the choice to choose among restaurants and their respective menus. The customer then places his order. After the order is placed he/she is asked if they want to keep choosing items or exit. Exiting this section takes them back to the main menu interface
- Admin panel: this section is designated for authorized personnel only. On entering this section one is first asked to verify that they are from the admin department, therefore they require to enter a password. Once that is done the admin can make amends to the menu items that are stored in form of linked lists. One can choose to add or delete menu items as necessary. Moreover, we can also choose to view the total sales for the day, and backup data in accordance to the respective day.
- About us: this simple function gives a brief description about us.
- Exit: exits the program.

Basic Flow of Program



Implementation using Linked lists

Using data structures, the organization of data has been done using a singly linked list. The singly linked list consists of a sequence of nodes which has a structure with the food name, price and quantity.

The data being organized in a linked list is the structure that contains the food details.

Have used the functions like insert front, insert end, delete a particular node, update the food item list and check out the cart items. Using two interfaces i.e customer and the admin panel, have learnt to develop an application that supports multi interfaces.

Chapter 3: Discussion on the Achievements

As we took on the task of coding the actual program we came across various challenges. As discussed earlier our main idea was creating the program using python but our lack of experience with the

language opted us to use C instead. Due to us only practicing small programs in class it was a true challenge creating a more complex program with a large numbers of functions and complex algorithms .On spending more time doing the project we realized it was rather difficult to implement other types of data structures other than just linked lists. To add further, we also planned on using circular queue to our program but seeing that it increased the level of complexity we had to start from scratch using linked list.

Though our idea was to use as many data structures as we could, we were not able to muster the time and skill to do so effectively, leaving us with just using linked lists in the end. Despite the time constraint, and the complexity of the project we were able to achieve a smooth running and highly effective program using just linked lists. It satisfied all the requirements we had planned for the program to have.

Chapter 4: Conclusion and Recommendation

In conclusion, the project was as we had wished for it to be, the exception being that we were able to use just one data structure in our program.

Future Enhancement

In future if we continue this project we can make a full fledged software, that can be used for any restaurants. We were thinking, by the end of 4th year we will launch this software for the local restaurants around KU. Now we were able to use only one of the many data structure, linked list. But as we gain more deep knowledge related to DSA we will use other Data Structure and make our little project more effective and more user friendly than now.

Reference

- Tutorialspoint.com
- Youtube.com