COMP 4

Tommy Tham

November 4, 2014

Contents

Chapter 1

Analysis

1.1 Introduction

1.1.1 Client Identification

My client is Linh Tham, the owner of Linh's Restaurant. Linh's Restaurant is a family run Chinese restaurant that is situated in small village called Fordham in Cambridgeshire. Linh works 'outside' usually on her own where she carries out many roles such as taking phone calls for orders/bookings, serving customers and calculating the bills. Outside is referred as the place where the the serving takes place. When times are busy, relatives come and help out at Linh's Restaurant.

Linh would like a more computerized system to be more efficient as manually transferring order details to the invoice book can be time consuming. In addition, when it is busy, using time more efficiently is definitely going to give a better service to the customers and would make it less stressful for Linh. Linh has basic knowledge on how to use a computer such as surfing the internet, checking emails and streaming videos.

1.1.2 Define the current system

Customers come in and get seated according to the number of people. Menus are given and the customers are asked what they would like to drink and what dishes they would like if they are ready. The dishes and drinks are recorded on seperate papers. The top copy of the ordering pad, where the dish order is recorded, is given to the chefs where they cook the dishes and one is kept for outside. Once the dishes are served, the customers are checked upon to see if there are any problems occasionally, and once the customers are satisfied and finish with their meal, they ask for the bill. The recorded order is then copied on to an invoice where the price is calculated for their meal. One invoice is kept and one is given to the customers. The meal is then paid and the customers leave.

The current system is paper based. A diary is used for bookings in which customers can book a table over the phone. A name, number of people on the table, a date and time are recorded in the diary. Orders are taken upfront which is recorded on an ordering pad where one copy is handed over to the kitchen and one is kept to refer to and to transfer order details onto an invoice form.

1.1.3 Describe the problems

When times are busy there could be confusion between on what has been ordered by what table. Also, having to rewrite the order into the invoice book takes time, this is a problem. Furthermore, any inexperienced workers will have to keep referring to the menu when taking orders or calculating the total bill to check if the dish is on the menu and the prices for each dish. This can also lead to a problem where the total price calculated is wrong. Additionally, recorded orders can go missing but that would only happen if the recorded order drops on the floor and no one realises it, this is something that is very unlikely to happen.

1.1.4 Section appendix

Interview with Linh Tham

What is the current system?

LT : I ask the customers what they would like to eat and drink and record it on an ordering pad, I then take top copy to the kitchen and keep the second copy for myself so I can refer to who ordered what. Once the customers has finished eating and ready to pay, I transfer all the details from the ordering pad on to the invoice form such as the drinks and dishes with the prices of each. I give a copy of the invoice to the customer and keep one for myself.

How are the second copies of the order and invoice created?

LT : Because of how thin the paper is on the ordering pad , writing things down marks down what I write on the second copy. However, the second copies are hard to read because the ink from the pen isn't exactly transferred.

What are the problems with the current way of doing things?

LT : Doing it manually is very time consuming as it takes one person just to rewrite everything on the invoice book. If that one person would be able to finish quicker, that person could help out which would benefit us.

What data or information is recorded in the current system?

LT : Food items, drinks,total price and the date of an order.

What are the benefits of the current system?

LT : As the system is paper based, any power cuts or weather issues, will not affect how we run the restaurant.

What should the new system be able to do?

LT : Having a way to look at what tables have ordered what, like a simulator, this will help the restaurant staff to keep track of tables and will reduce confusion. Storing sit down orders would be helpful also.

Having a way to look at what tables have ordered what, like a simulator, this will help the restaurant staff to keep track of tables and will reduce confusion. Storing sit down orders would be helpful also.

Would you like to store phone call orders?

LT: No, I would only like to store sit down orders.

How long would you like to store the information?

LT: I would like to store the information for 3 months.

1.2 Investigation

1.2.1 The current system

Data sources and destinations

There are two main data sources in the current system, the menu and the customer. The menu contains foods and drinks the customer can choose from, the restaurant staff takes the order and then writes down the drinks and dishes onto seperate ordering pads without prices. The details of the order is copied onto the invoice including prices and the date once the customer is finished. Each dish ordered is recorded on the invoice however, each drink ordered isnt and so the total price of drinks ordered is recorded instead and referred as 'Drinks' on the invoice form. Additionally, the number of people on the table isn't recorded on the invoice.

Source	**Data**	**Example Data**	**Destination**
Menu	Drink and dishes with prices	Orange Juice £0.70 Special fried rice £3.70	Customer
Customer	Drink	Bottled water	Restaurant staff
Customer	Dish	Wonton soup	Restaurant staff
Restaurant staff	Drink ordered by customer, table number	Bottled water Sprite Table No. 3	Ordering pad 1
Restaurant staff	Dish ordered by customer, date of order, number of people, table number	Wonton soup Special fried rice 30/9/14 Covers 2 Table No. 3	Ordering pad 2
Ordering pad 1	Total price of drinks - each drink is not specified on invoice, table number	(£0.60+£0.70) Total £1.30 Table No. 3	Invoice pad
Ordering pad 2	Dishes ordered by customer including price of each dish, table number	Wonton soup £1.80 Special fried rice £3.70 Table No. 3	Invoice pad
Restaurant staff	Total price of order	Total price £6.8	Invoice pad
Invoice pad	Copy of invoice	Wonton soup £1.70 Special fried rice £3.70 Drinks £1.30 Total price £6.8 Date 30/9/14 Table No. 3	Customer

Algorithms

Algorithm 1 Taking an order

```
SET OrderTaken TO false

WHILE notOrderTaken DO
    IF Customer ready to order THEN
        RECEIVE Order FROM KEYBOARD
        SET OrderTaken TO true
    ELSE
Wait
    END IF
END WHILE
```

Algorithm 2 Generating invoice

```
SET InvoiceGenerated TO false

WHILE notInvoiceGenerated DO
    IF Customer has finished ordering THEN
Copy order details from order pad onto invoice pad
Get prices of each dish and drink ordered from menu
Copy prices onto invoice pad
Calculate total price
Add date
        SET InvoiceGenerated TO true
    ELSE
Wait for customer to ask for the bill
    END IF
END WHILE
```

Algorithm 3 Payment

```
SET Payment TO false

WHILE notPayment DO
    IF Customer ask for bill THEN
Give invoice
        RECEIVE Payment FROM KEYBOARD
        SET Payment TO true
    ELSE
Wait
    END IF
END WHILE
```

Data flow diagram

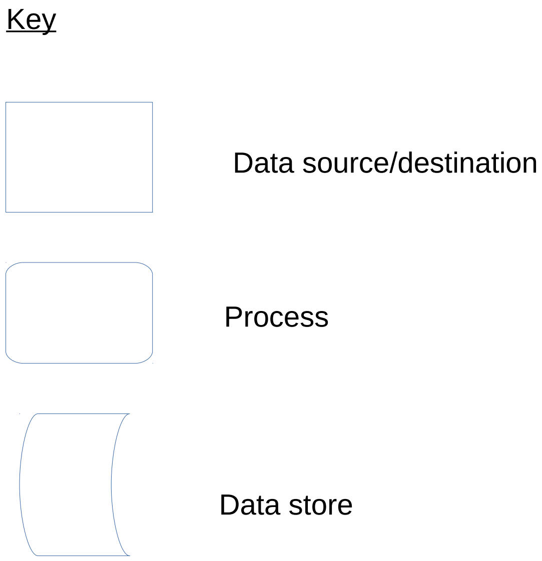


Figure 1.1: Data flow key

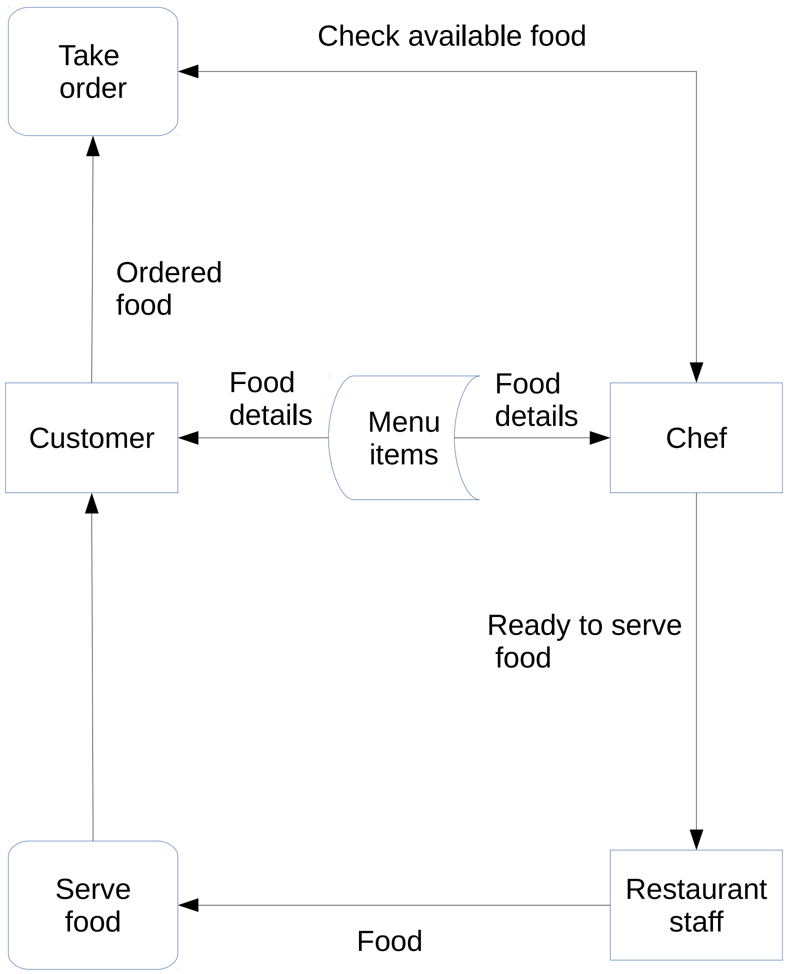


Figure 1.2: Data flow diagram of placing an order

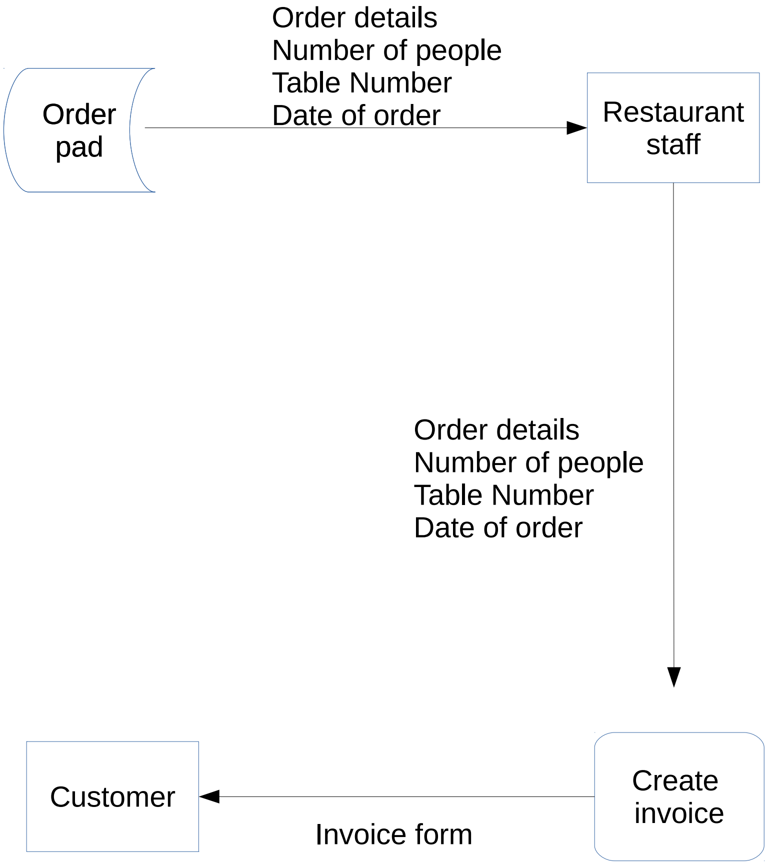
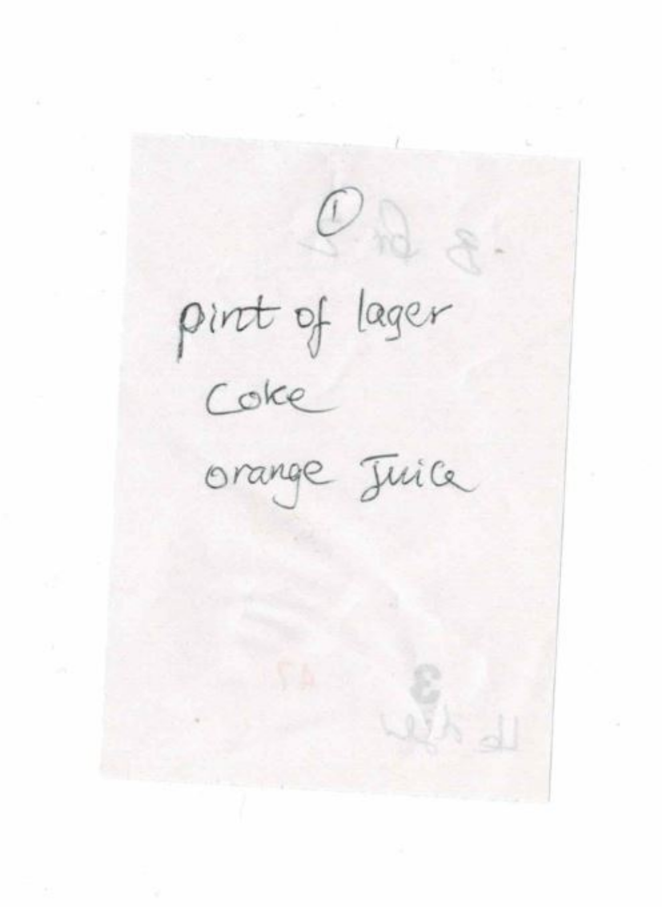


Figure 1.3: Data flow diagram of generating an invoice

Input Forms, Output Forms, Report Formats

Drinks are recorded seperately from dishes as shown below. The number at the top represents what table number this order is from.



1

pint of lager

Coke

orange Juice

Figure 1.4: Writing down drinks ordered on the drink pad

Below is an example of what the ordering pad looks like when a customer's order has been taken. It provides information about the order such as the table number, how many people is seated, dishes ordered and the date the order has taken place. Two copies of this is made, one is taken to the chefs and one is kept for the waitors. This is an input form.

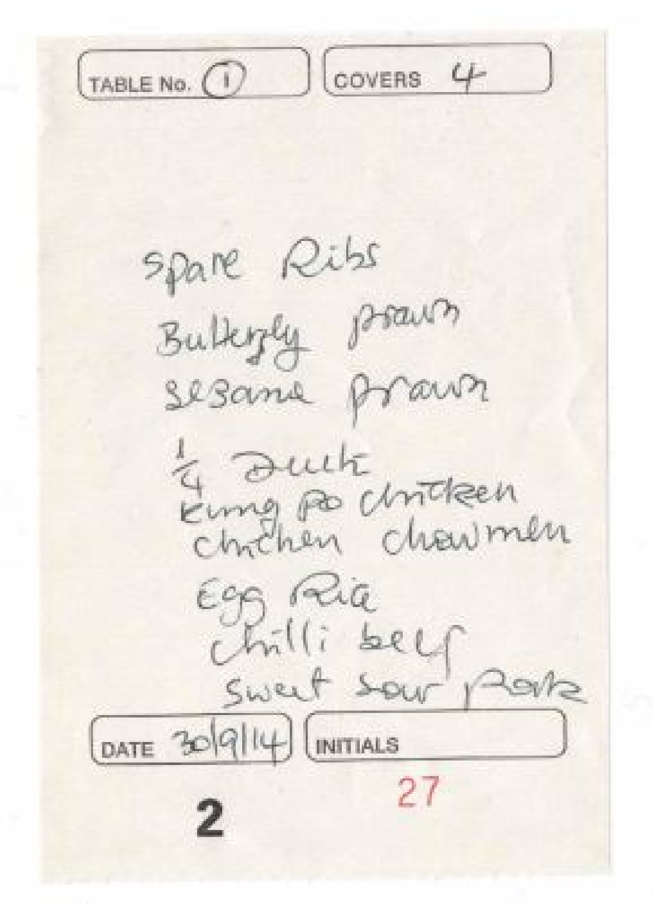


TABLE No. 1 COVERS 4

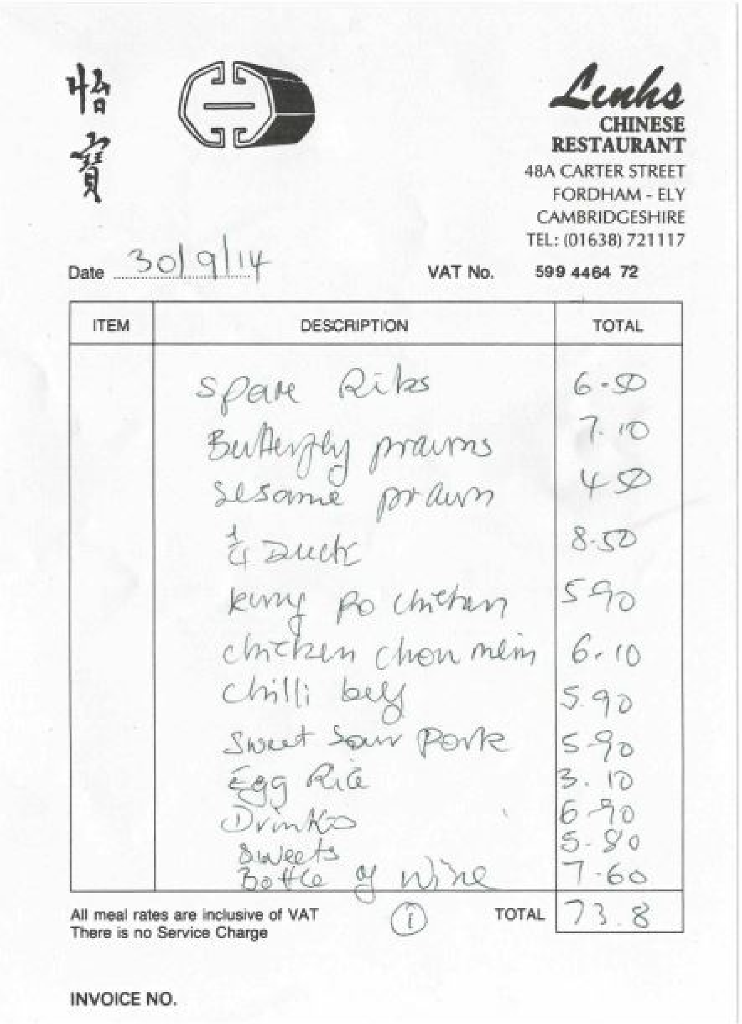
spare Ribs
Butterfly prawn
sesame prawn
$\frac{1}{4}$ Duck
kung po chicken
chicken chowmen
Egg Rice
Chilli beef
sweet sour pork

DATE 30/9/14 INITIALS

2 27

Figure 1.5: Getting an order from a customer

A picture of an invoice is shown below, the information has been transferred from the ordering pad, as shown above, to the invoice pad. An invoice is created once a table has finished eating and ready to pay. Only the date, description, prices and total price is put on the invoice. This is an output which is given to the customers and another copy of the invoice is kept.



怡寶

Lenhs
CHINESE
RESTAURANT
48A CARTER STREET
FORDHAM - ELY
CAMBRIDGESHIRE
TEL: (01638) 721117

Date 30/9/14 VAT No. 599 4464 72

ITEM	DESCRIPTION	TOTAL
	Spare Ribs	6.50
	Butterfly prawns	7.10
	Sesame prawn	4.50
	1/4 Duck	8.50
	Kung Po chicken	5.90
	chicken chow mein	6.10
	Chilli beef	5.90
	Sweet Sour Pork	5.90
	Egg Rice	3.10
	Drinks	6.90
	Sweets	5.80
	Bottle of Wine	7.60
	1 TOTAL	73.8

All meal rates are inclusive of VAT
There is no Service Charge

INVOICE NO.

Figure 1.6: Creating invoice

1.2.2 The proposed system

Data sources and destinations

In the proposed system, getting an order from the customer is still the same via using restaurant staff and an order pad. The only change in the propose system is transferring the order details onto an invoice.

Source	**Data**	**Example Data**	**Destination**
Menu	Dish and drink	Spare ribs, orange juice	Customer
Customer	Drink ordered	Orange juice	Restaurant staff
Customer	Dish ordered	Wonton soup	Restaurant staff
Restaurant staff	Drink ordered by customer, table number	Orange juice Table No. 1	Ordering pad
Restaurant staff	Dish ordered by customer, table number, number of people, date of order	Wonton soup, Table No. 1, Covers 1 04/09/14	Ordering pad
Proposed system software	Invoice form	04/09/14 Wontop soup £ 1.80 Drinks £0.7 Total price £2.50	Customer

The new part of the system's data sources and destinations is shown below. Entering the food item onto the software should automatically retrieve its price from the menu database. After a customer has finished with their meal, the simulator saves the Table status (drinks, dishes, table number and date) to the order history database and creates an invoice form.

Source	**Data**	**Data type**	**Destination**
Restaurant staff	Dish	String	Computer - Table status
Restaurant staff	Drink	String	Computer - Table status
Restaurant staff	TableNumber	Integer	Computer - Table status
Restaurant staff	NumberOfPeople	Integer	Computer - Table status
Restaurant staff	DateOfOrder	Date	Computer - Table status
Computer - Table status	OrderID	Integer	Database - Order records
Computer - Table status	Dish	String	Database - Order records
Computer - Table status	Drink	String	Database - Order records
Computer - Table status	TableNumber	Integer	Database - Order records
Computer - Table status	DateOfOrder	Date	Database - Order records
Computer - Table status	TotalDrinkPrice	Float	Database - Order records
Computer - Table status	TotalPrice (TotalDrinkPrice + each dish)	Float	Database - Order records
Computer - Table status	InvoiceForm	string	InvoiceFolder

Data flow diagram

The data flow diagram of placing an order will be the same due to no changes to the way of placing and processing the order.

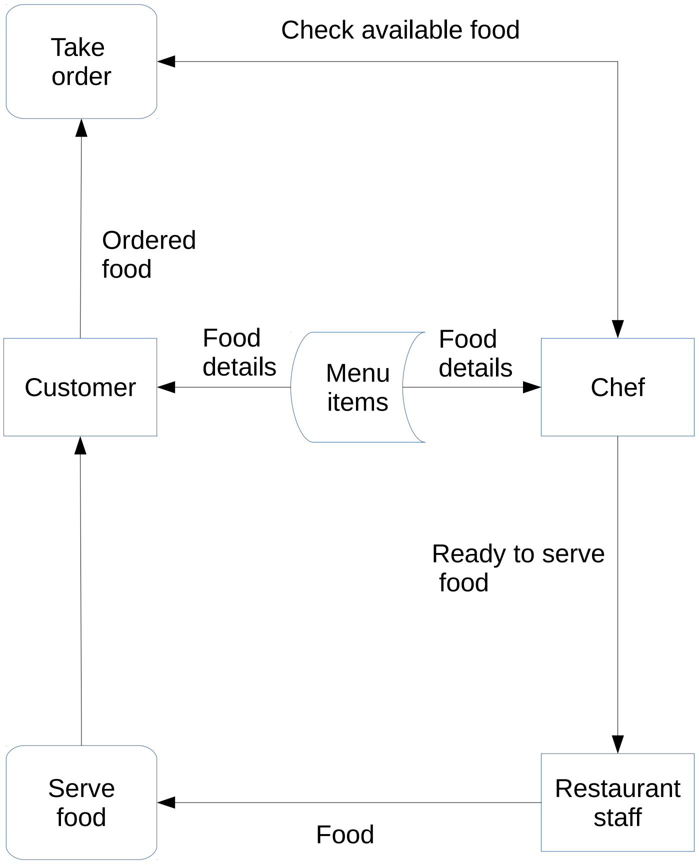


Figure 1.7: A data flow diagram of the proposed system - placing and processing the order

The proposed system will make the restaurant staff input data into the system which will be shown on the application if the user checks what table has ordered what. In addition the inputed data saved in a database once the customer has finished with their meal. Also, invoices will be created though this application.

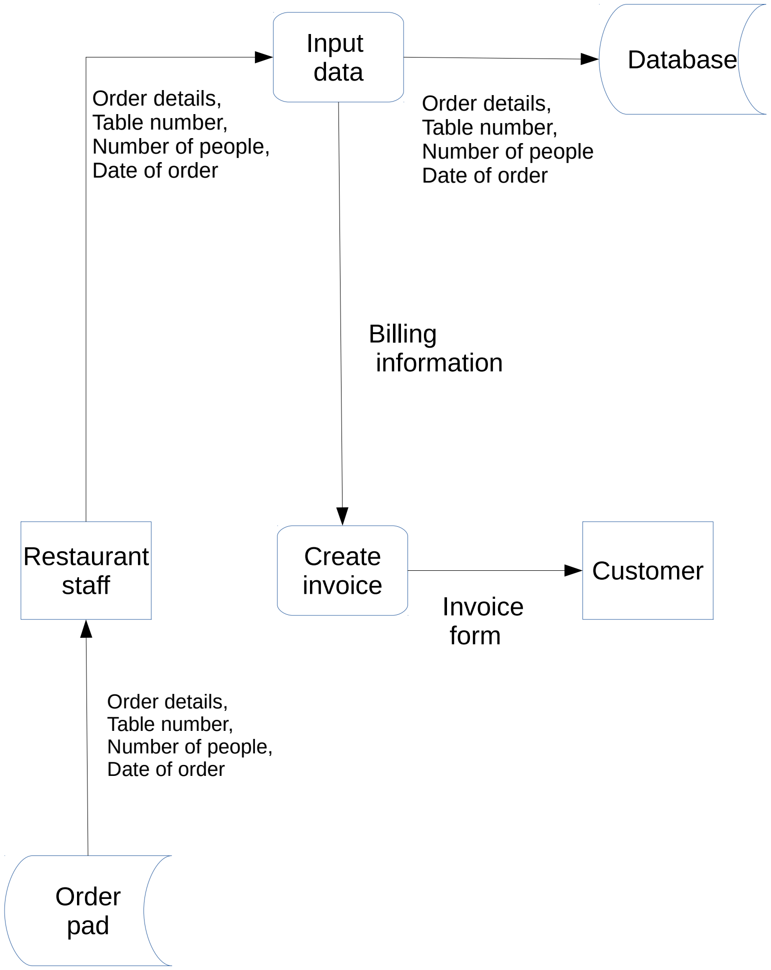


Figure 1.8: Data flow diagram proposed system

Data dictionary

Name	**Data Type**	**Length**	**Validation**	**Example Data**
TableNumber	Integer	1 - 16	Range	13
NumberOfPeople	Integer	1 - 20	Range	4
MenuItem	String	1 - 20 Characters	Length	Spare ribs
ItemQuantity	Integer	1 - 10	Range	4
ItemPrice	Float	0 - 20	Range	3.2
TotalPrice	Float	0 - 500	Range	54.4
DateOfOrder	Date	4 - 6	Format	16/11/14
InvoiceCreated	Boolean		Presence Check	

Volumetrics

As an ascii character is 1 byte, there will be 35 bytes for one sitdown order. 35*30(approximately the max sit down orders per day) = 1050 bytes is stored per day. Linh's restaurant is open 6 days a week so 1050*6 = 6300 bytes and Linh has stated that she would like to store the information for 3 months so 6300*13.2(weeks) = 83160 bytes will need to be stored.

81360 bytes is equivalent to 79.45 kilobytes (81360/1024). 79.45 kb would be needed to store 3 months of information. The software it self will contain pictures which will increase the size by roughly 2MB. Therefore the total space required would be 6MB if the application itself took 4MB without any images (2MB + 4MB).

1.3 Objectives

1.3.1 General Objectives

- Create a restaurant simulator to track orders
- Simple and clear GUI for user-friendly experience.
- Having the ability to easily modify orders.
- Create a digital invoice after table has finished their meal.
- Storing orders.

1.3.2 Specific Objectives

Simple and clear GUI

- Having a very simple birds eye view image of the restaurant which is made out of shapes to ease the understanding of where each table is.
- Label table with their corresponding number.
- Table shapes will be big so it won't be hard to click on them but not so big that 16 tables can fit on the GUI.
- Clicking on table will bring up a window which shows the status such as the date and food items ordered with noticeable order modification options.

Order alterations

- Have clear Add, Delete and Create invoice buttons.
- When user chooses the add option, have an input box appear where user can type in an ID for a dish/drink or the actual name of the dish/drink.
- Make the input search function not case sensitive.
- When user wants to delete a food item off the list, have clear red X boxes appear next to the name. When red X boxes are clicked on and with confirmation, the item gets deleted.
- Have an up arrow or bottom arrow button just in case a customer orders another food item which is already on the list. The up arrow would increase the quantity of the item by 1 and the down arrow would decrease the item by 1 .
- Clicking on create invoice button will clear the information on the table status and save the digital invoice in a folder.

Track orders

- Drinks and dishes will be seperated by columns.
- Clicking on a table will bring up a small window with the list of food items that the table has ordered, formatted like the invoice form shown on page 15. This also includes the date and table number.

Invoice creation

- Automatically creating a digital invoice when a customer has finished.
- Calculate total price
- The digital invoice will look very similar to the invoice on page 15.
- Invoice will contain the items ordered, prices of each and total price.

- Have the option to print out invoice.

Storing orders

- When using the clear information button, the information is stored in the database.
- Filtering database for user if searching specific information.
- Have an option to view database.

1.3.3 Core Objectives

- Have a working simulator that will have the restaurant layout
- Having clickable tables that will bring up a window showing a digital invoice
- The digital invoice will show the current status such as items ordered, date of order and number of people on the table.
- Application must be able to modify orders
- Application must be able to generate an invoice after table has finished with their meal

1.3.4 Other Objectives

- Print invoice function
- Store order data in a database
- Database search functions such as sort and filtering.

1.4 ER Diagrams and Descriptions

1.4.1 ER Diagram

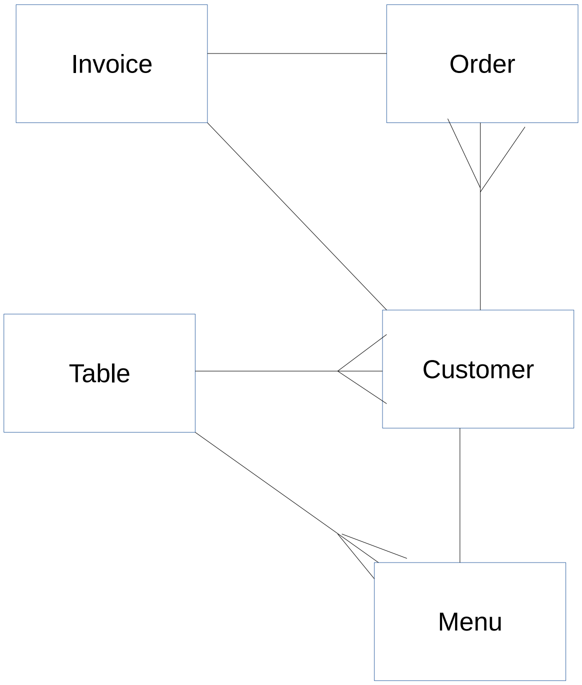


Figure 1.9: E-R Diagram

1.4.2 Entity Descriptions

Customer(CustomerID, *TableID*, *OrderID*, NumberOfPeople, Invoice, Date)

Order(OrderID,*CustomerID*,*TableID*, *MenuID*, DishOrdered, DrinkOrdered, Quantity)

Table(TableID, *OrderID*, *CustomerID*, TableNumber)

Menu(MenuID, Dishes, Drinks, DishPrice, DrinkPrice)

Invoice(InvoiceID, *CustomerID*, *OrderID*, TotalDrinkPrice, TotalPrice)

1.5 Object Analysis

1.5.1 Object Listing

- Customer
- RestaurantStaff
- Dish
- Drink
- Invoice
- Menu

1.5.2 Relationship diagrams

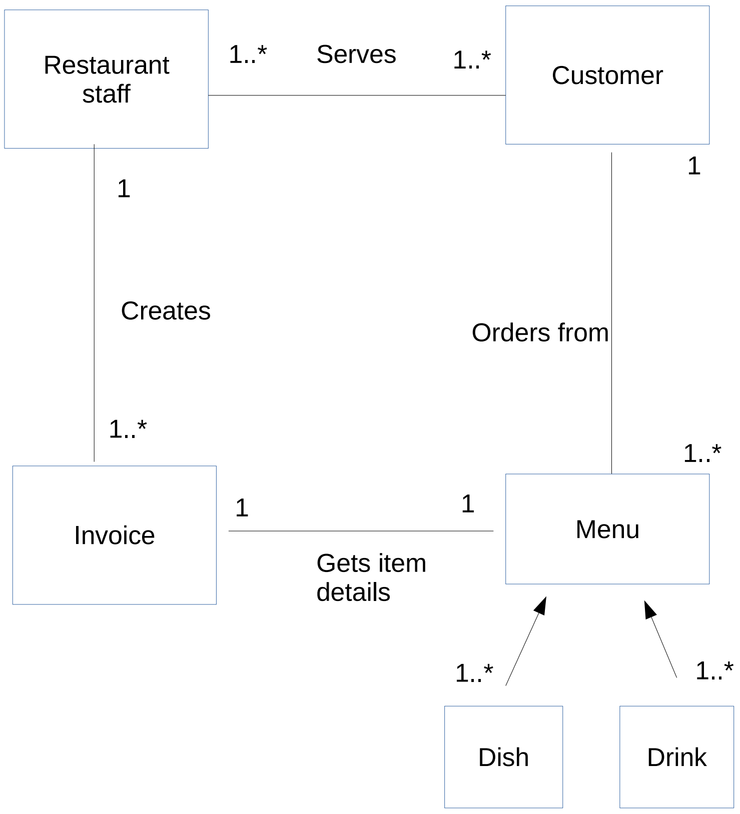


Figure 1.10: Relationship diagram

1.5.3 Class definitions

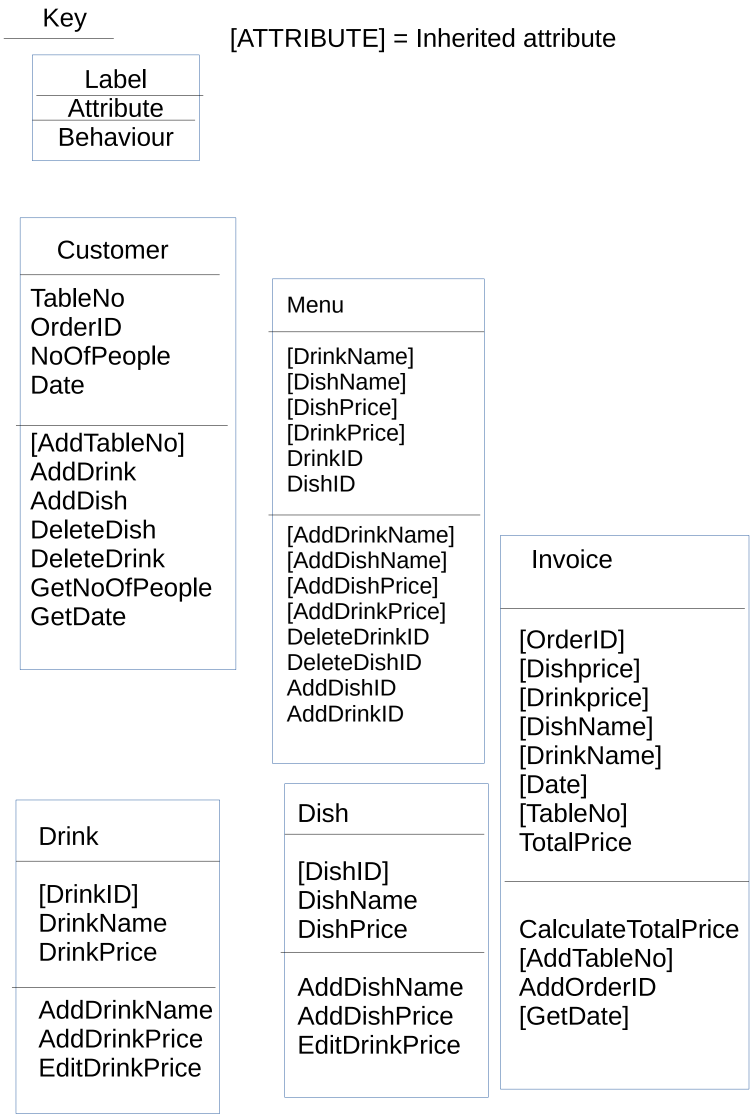


Figure 1.11: Class diagram

1.6 Other Abstractions and Graphs

1.7 Constraints

1.7.1 Hardware

The current computer specifications is as follows:

- 19” Display
- AMD FX(fm) - 6300 six-core CPU 3.50Hz
- 8GB RAM
- NViDiA GeForce 9600 GT 1GB
- Windows 8.1 64 bit

There shouldn’t be any constraints apart from the fact that the new system will have to be designed to fit the 19” screen. Also the position of where the computer will be placed in the restaurant is a limitation.

1.7.2 Software

The current computer uses Windows 8.1 and Linh would prefer it to stay that way as she is familiar with the operating system. This is not a problem as the proposed system will run fine on Windows 8.1. Apart from that, Linh has not stated what software can or cannot be used.

1.7.3 Time

Linh has not set me a deadline for the new system and is in no rush for it to get done. Therefore the deadline will be Friday 27th March 2015 which is the coursework deadline set by my teacher.

1.7.4 User Knowledge

Linh has basic knowledge on how to use a computer such as being able to check emails and simple web surfing. Basic knowledge will not constrain the project as one of the objectives is for the software to be simple and clear.

1.7.5 Access restrictions

All working staff should be able to use this software due to the nature on how the business is run. All waiting staff should be able to carry out the same roles such taking an order, serving and creating an invoice form. However, customers should not be able to access this application at all which could be considered as a constrait. A simple enter password-to-access mechanic could be used as a solution to this.

1.8 Limitations

1.8.1 Areas which will not be included in computerisation

The method of taking orders will not be computerised as it more convenient to just take orders by pad. Using an ordering pad is useful as it is small, light and easy to carry around. Also, the payment system (receiving money and giving back change) will not be computerised as there are no problems with the current payment system. More problems will likely be created if it was to be computerised such as giving back the correct amount of change and registering the amount of money received.

1.8.2 Areas considered for future computerisation

Tracking bookings for tables can be a feature for later as it could be helpful if the book of table bookings goes missing or if theres no more space to write down bookings. In addition, Linh has not stated that she wanted take aways to be computerised. This could be an additional feature in the future to the program where it creates invoices for take aways.

1.9 Solutions

1.9.1 Alternative solutions

Solution	**Advantages**	**Disadvantages**
Python Desktop Application with a GUI	The design can be changed according to client needs. Not complicated to use. Very low cost. User-friendly and problems with current system will be fixed. Extra features can be implemented.	Application will take up noticeable computer storage. Will take a long time to create GUI application. If theres a power cut then system will not be useable
Touch sceen self-order system	Customer has more freedom. Less work for restaurant staff. Problems with current system will be gone.	More hardware and software needed - can be very costly. Technical issues will be hard to fix.
Getting someone to do invoices only	Will solve the main problem with current system. No need for a computer.	Will be hard to find someone who will only do invoices. If business isn't busy then invoice person will be almost useless. Could be more costly in long run.
Redesign current manual system	No cost or very low cost as no computer/software will be needed. Current manual system is simple.	May not be able to fix problems. Will take some time to figure out how to fix problem.

1.9.2 Justification of chosen solution

I have chosen Python Desktop Application with a GUI as the solution because of many reasons. One reason is that the touch screen solution will be very costly and customers would have to queue up to use the machine if it gets busy. This will affect how the business will run as many customers do not like to wait. Also, hiring out someone to do invoices will not be efficient as money will be wasted if business is not busy. Furthermore redesigning the current system will take time as Linh would need to figure how fix the problems in the system and also this will most likely not fix most problems with the current system. Therefore I choice Python Desktop Application because it would not need any further hardware, this will not negatively affect customers experience at the restaurant in any way and due to Python being very flexible, the program can always be changed to Linh's wants.

Chapter 2

Design

2.1 Overall System Design

2.1.1 Short description of the main parts of the system

2.1.2 System flowcharts showing an overview of the complete system

2.2 User Interface Designs

2.3 Program Structure

2.3.1 Top-down design structure charts

2.3.2 Algorithms in pseudo-code for each data transformation process

2.3.3 Object Diagrams

2.3.4 Class Definitions

2.4 Prototyping

2.5 Definition of Data Requirements

2.5.1 Identification of all data input items

2.5.2 Identification of all data output items

2.5.3 Explanation of how data output items are generated

2.5.4 Data Dictionary

2.5.5 Identification of appropriate storage media

2.7 Security and Integrity of the System and Data

2.7.1 Security and Integrity of Data

2.7.2 System Security

2.8 Validation

2.9 Testing

2.9.1 Outline Plan

Test Series	Purpose of Test Series	Testing Strategy	Strategy Rationale
Example	Example	Example	Example

2.9.2 Detailed Plan

Test Series	Purpose of Test	Test Description	Test Data	Test Data Type (Normal/ Erroneous/ Boundary)	Expected Result	Actual Result	Evidence
Example	Example	Example	Example	Example	Example	Example	Example

Chapter 3

Testing

3.1 Test Plan

3.1.1 Original Outline Plan

Test Series	Purpose of Test Series	Testing Strategy	Strategy Rationale
Example	Example	Example	Example

3.1.2 Changes to Outline Plan

Test Series	Purpose of Test Series	Testing Strategy	Strategy Rationale
Example	Example	Example	Example

3.1.3 Original Detailed Plan

Test Series	Purpose of Test	Test Description	Test Data	Test Data Type (Normal/ Erroneous/ Boundary)	Expected Result	Actual Result	Evidence
Example	Example	Example	Example	Example	Example	Example	Example

3.1.4 Changes to Detailed Plan

Test Series	**Purpose of Test**	**Test Description**	**Test Data**	**Test Data Type (Normal/ Erroneous/ Boundary)**	**Expected Result**	**Actual Result**	**Evidence**
Example	Example	Example	Example	Example	Example	Example	Example

3.2 Test Data

3.2.1 Original Test Data

3.2.2 Changes to Test Data

3.3 Annotated Samples

3.3.1 Actual Results

3.3.2 Evidence

3.4 Evaluation

3.4.1 Approach to Testing

3.4.2 Problems Encountered

3.4.3 Strengths of Testing

3.4.4 Weaknesses of Testing

3.4.5 Reliability of Application

3.4.6 Robustness of Application

Chapter 4

System Maintenance

4.1 Environment

4.1.1 Software

4.1.2 Usage Explanation

4.1.3 Features Used

4.2 System Overview

4.2.1 System Component

4.3 Code Structure

4.3.1 Particular Code Section

4.4 Variable Listing

4.5 System Evidence

4.5.1 User Interface

4.5.2 ER Diagram

4.5.3 Database Table Views

4.5.4 Database SQL

4.5.5 SQL Queries

4.6 Testing

4.6.1 Summary of Results

4.10.1 Module 1

Chapter 5

User Manual

5.1 Introduction

5.2 Installation

5.2.1 Prerequisite Installation

Installing Python

Installing PyQt

Etc.

5.2.2 System Installation

5.2.3 Running the System

5.3 Tutorial

5.3.1 Introduction

5.3.2 Assumptions

5.3.3 Tutorial Questions

Question 1

Question 2

5.3.4 Saving

5.3.5 Limitations

5.4 Error Recovery

5.4.1 Error 1

Chapter 6

Evaluation

6.1 Customer Requirements

6.1.1 Objective Evaluation

6.2 Effectiveness

6.2.1 Objective Evaluation

6.3 Learnability

6.4 Usability

6.5 Maintainability

6.6 Suggestions for Improvement

6.7 End User Evidence

6.7.1 Questionnaires

6.7.2 Graphs

6.7.3 Written Statements