Criterion B: Design

Overall Structure

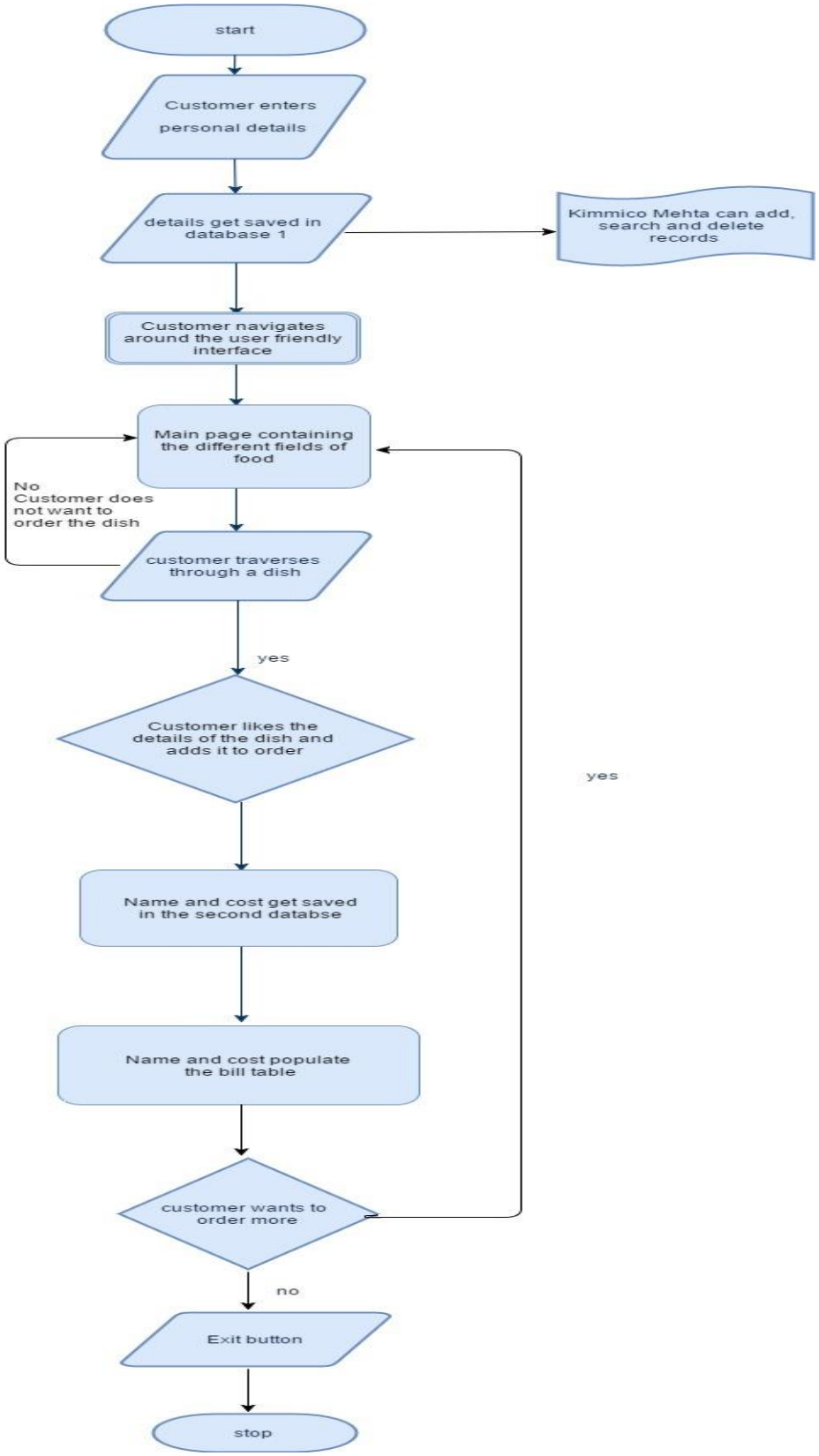

Figure 1: Algorithm for the working of the software

Internal Structure

In order to gain customer information (seen in the first node in algorithm 1), a cover page would need to be created in the software which would request for certain customer details. The design of the cover page is seen in the image below:

Figure 2: Cover Page

This cover-page would be the first JFrame form displayed to the customer. Without the filling of this form the customer would not be able to move forward and place an order. In order to ensure that the customer enters correct details, validation checks need to take place as shown in table 1.

Table 1: Validation checks for cover page	
First Name	Presence check
Surname	Presence check
Email id	Presence check
Table number	Presence check and Range check
Birthday	Presence check

These details should then be stored in the external flat-file database, Microsoft Office Access with the following data types in table 2:

Table 2: Data Type			
Field name	**Data type**	**Description**	**Other information**
ID	Integer	Uniquely identifies each customer	Serves as a primary key in order to differentiate each customer.
First Name	Text	First name of customer	
Surname	Text	Family name of customer	
Table Number	Integer	Number of the table the customer is seated on	
Email Id	Text	Customer's email id	
Birthday	Date	They day the customer was born	

The birthday of the customer is requested in order to send birthday emails from minus 18 to give a personalized touch.

Moving on, the details of each dish needs to be presented to the user. Each dish should be viewed with the ingredients which were used, along with the picture of the dish. This helps in ensuring maximum customer knowledge. The image below displays a layout for each dish:

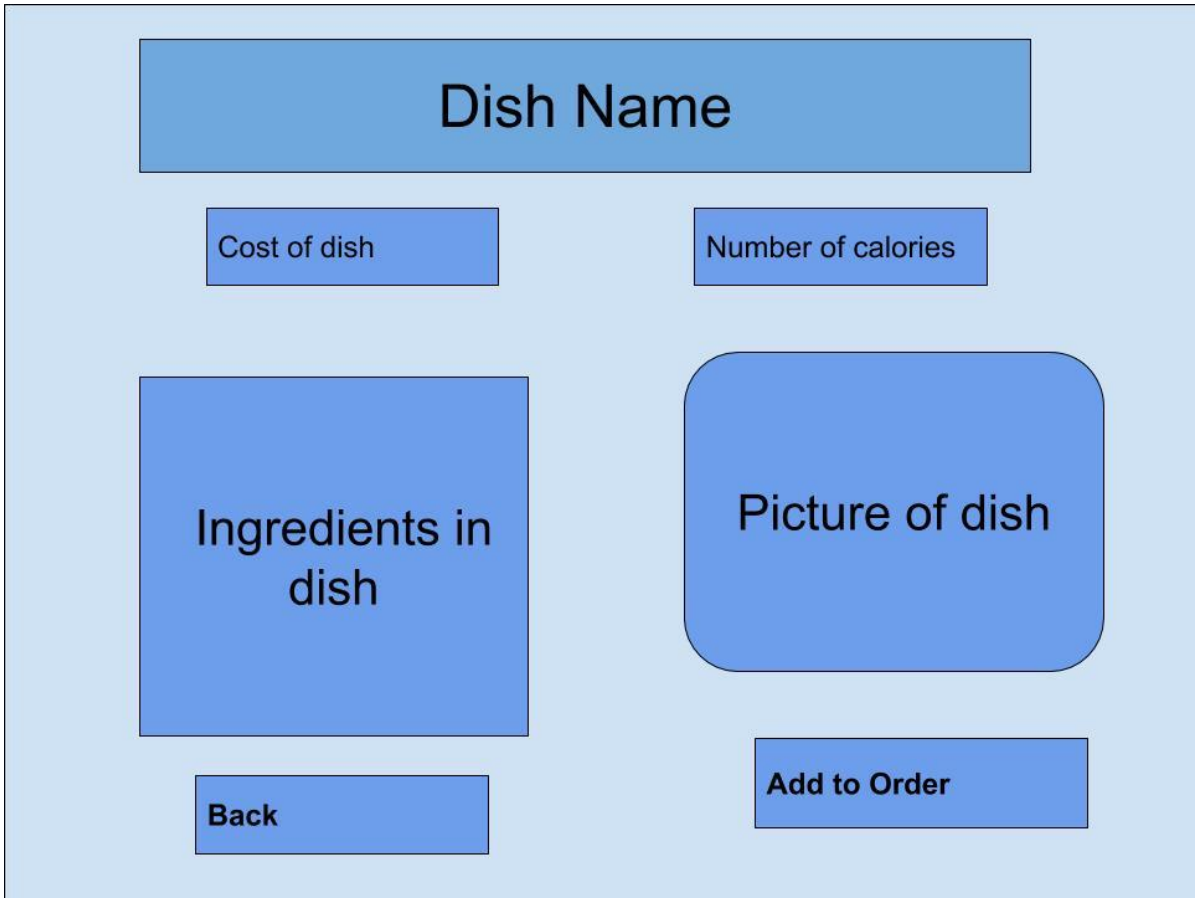

Figure 3: layout for each dish

Buttons such as “back” and “add to order” should be present to transfer from one jframe to another. Once a customer orders a dish, then the dish details should be added to the bill. In order to create the bill, the details of the dish need be stored in another Access database. The fields and data types of this second database are shown in table 3:

Table 3: Order details in Access database			
Field Name	**Data type**	**Description**	**Other information**
ID	Integer	Uniquely identified each dish	Serves as a primary key
Dish	Text	Name of the dish which is ordered by the customer	
Cost	Integer	The price of the dish ordered	

The order details should then populate a table in the Jframe producing a bill. The layout of this bill can be seen in the image below:

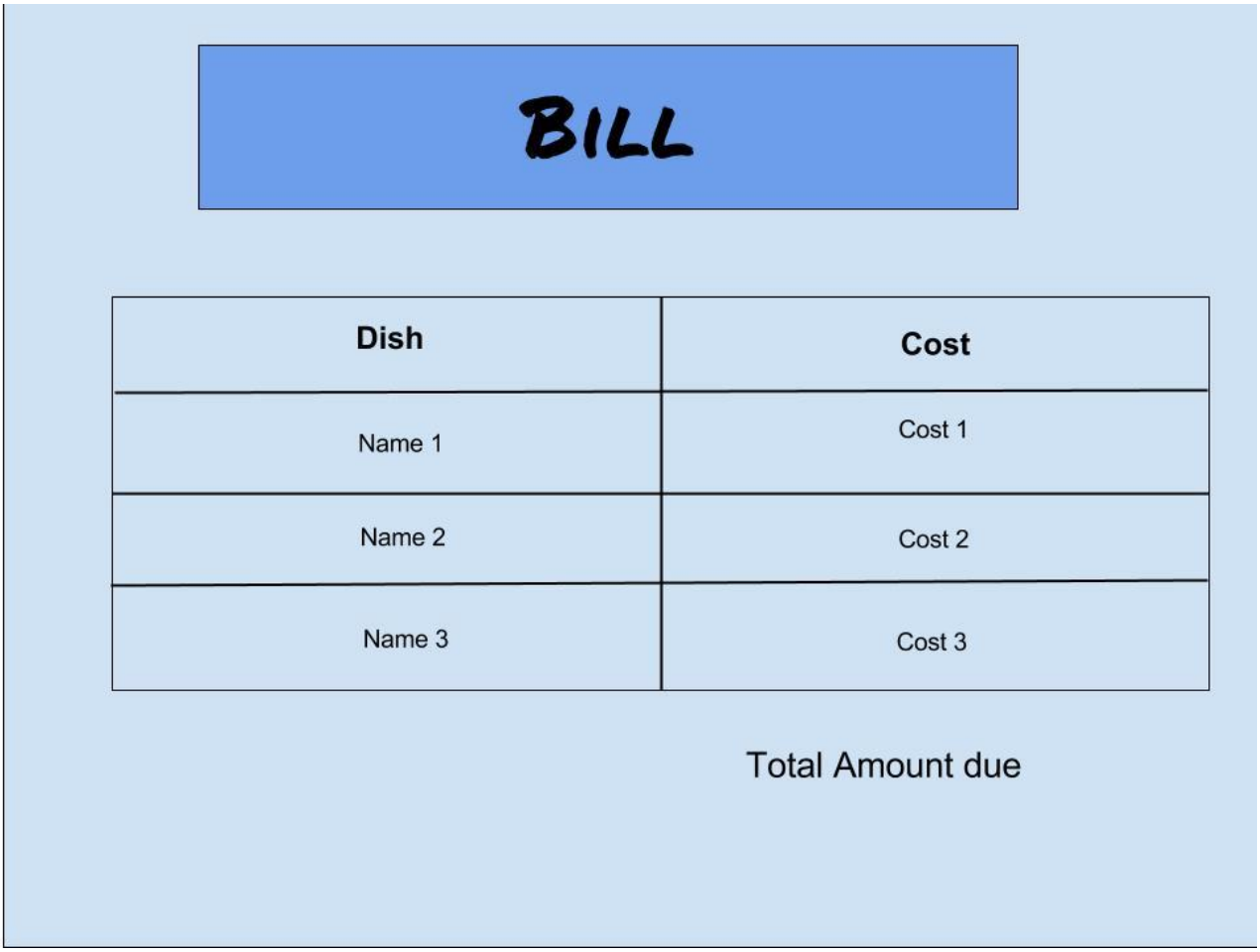

Figure 4: layout of bill

Test Plan

For Database 1: customer details			
Test Type	**Nature of Test**	**Example**	**Outcome**
Test with normal data for all five fields	The customer can move forward and see the menu.	First Name: Shalaka Surname: Mehta Table number: 4 Email id: shal.m@gmail.com Birthday: 2nd September 1998	Data accepted and stored in the database.
Test with extreme data of table number	The customer can move forward and see the menu.	Table numbers range from 1-10 Table number: 1 Table number: 10	Data accepted and stored in the database.
Test with abnormal data of table number and birthday	The customer should not be able to move forward and see the menu.	Table number: 100 Birthday: 30th February 2019	Error message is given to the user, and is prompted to enter valid data.
Test with inappropriate data such as leaving fields empty	The customer should not be able to move forward and see the menu.	Table number: First Name: Surname: Email ID:	Error message is given to the user, and is prompted to enter data.
Adding records in database	Kimmico Mehta should be able to add records	First Name: Sarah Last Name: Davis Email id: sarah.da@gmail.com	Data is valid, client should be able to add record.

Deleting records in database	Kimmico Mehta should be able to delete certain record	Delete this record: First Name: Serena Last Name: Adams	If this data is present in database, client can delete the record.
For the software			
Test Type	**Nature of Test**	**Example**	**Outcome**
Button functions	Whenever button is clicked should switch Jframe's	Back button Add to Order button	Open's another page, through the use of creating objects.
Bill production	Every time a dish is ordered it should be shown in the bill table	Add order pesto penne The name and cost should appear in the bill table	Data should be stored in database 2 containing the order and then fetched into table.
Total amount function	The total amount should be calculated from all the costs of dishes ordered	Sandwich: 230 Coffee: 300 Total Amount: 530	The total amount should be displayed since figures are accurate.
Close the program	Should exit the program	Exit button	Software should shut down

For Database 2: Order details			
Test Type	**Nature of Test**	**Example**	**Outcome**
Records of dish should be seen in database	Every time add to Order is pressed the record should be seen in the database	Add to Order	If the data matches the data type in database, records should be added.