Restaurant Management System

Technical Documentation

March’13

Developed By:

Group 2

B. Nitin Chandra (11010109)

Shubham Luhadia (11010176)

Nishant Yadav (11010147)

Sachin Aglave (11010160)

Madhuri Tikhe (11010133)

Mamidi Prashant (11010135)

Table of Contents

Project Goals

To develop a Restaurant Management System

in Visual C++ 2010 with basic services:

1) Shows the menu for customers to place orders, with an option to specify whether the customer wants to dine – in, take – out or have a home delivery. Extra charges for packing and home delivery.
2) Calculates Taxes.
3) Checks the frequency of a customer in the restaurant the special Customer Cards issued for this purpose. If a customer is very frequent in the restaurant, offers discounts.
4) Generates elaborate bills and maintains customer records.

Introduction

Restaurant Management System is software developed in Visual C++ 2010 using the Microsoft .Net Framework.

It uses MySQL , the open source relational database management system (RDBMS) for maintaining the databases using the phpMyAdmin module of XAMPP package to graphically administer MySQL databases and visually design database structures.

The software is a Windows Form Application consisting of 28 windows forms. Except of the database table phasedecide and those related to predefined menus, all the tables are generated dynamically from the program itself as and when needed.

Modules

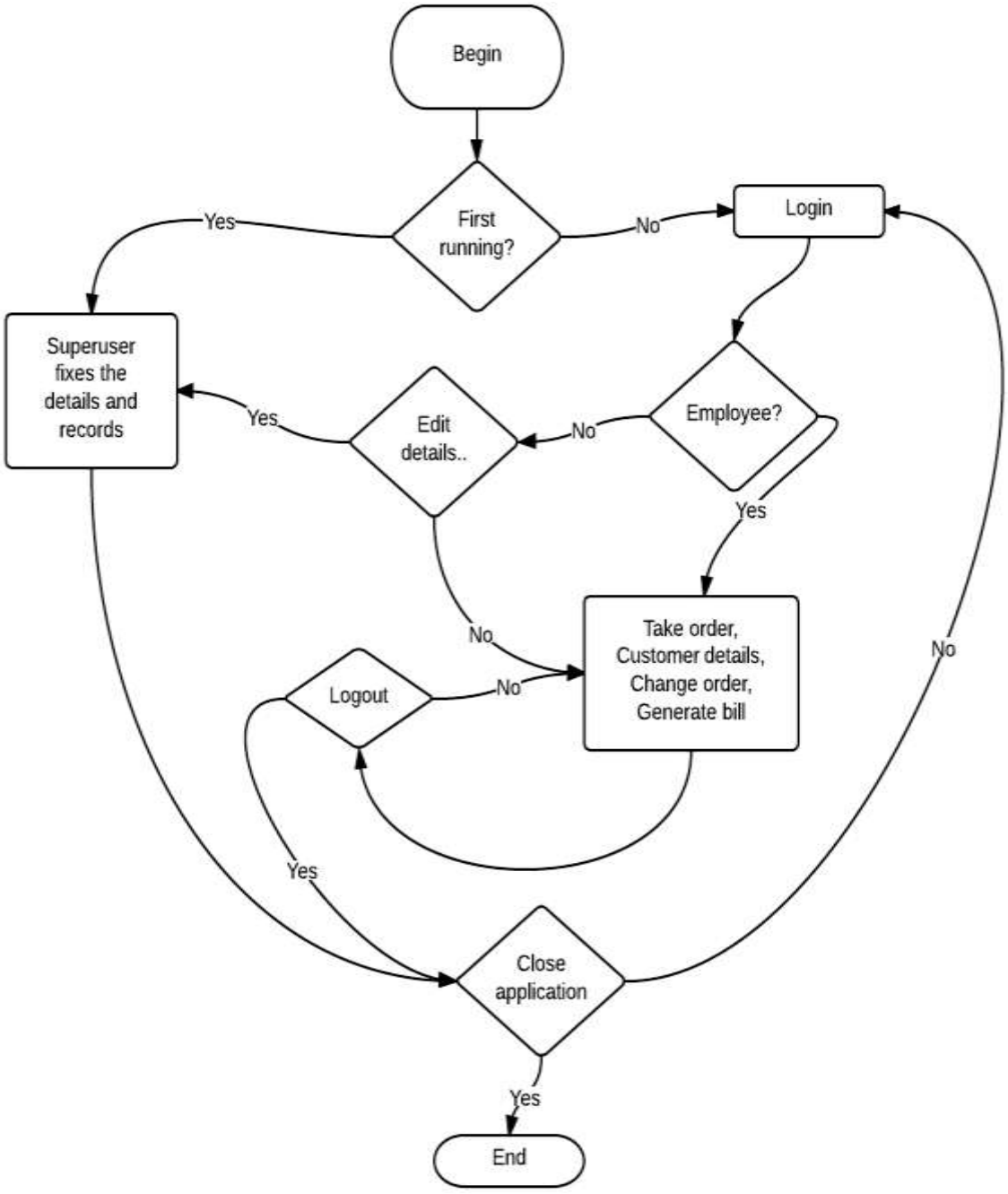
The whole software is partitioned into two broad modules:

1) Installation phase: Here the super user fixes all the details of the restaurant over a sequence of 10 windows forms and all the database tables are created and initialized.
2) Daily Phase: Here any employee (including the super user) logs in and if (s) he is found to be a valid employee, then (s) he can perform the various operations of :
    a) Placing Order: Customer selects the food items from the menu and the order is stored in the database.
    b) Changing Order: Customer can delete an ordered food item within the predefined maximum cancellation time. Adding items can be done at any point of time when the restaurant is open
    c) Generating Bill: Generate an elaborate bill including calculation of sub total, discounts and taxes. Option to print available.

d) Issuing Customer Cards: If a customer card is willing, his information can be stored in the database to track his/her frequency in return of some discounts set by the super user.

*All the database information is available under Database Tables (pg.)

Flowchart representing the modules



Installation Phase

1) **Form0**: When the software is run the start form appears. This form is set as a splash screen. When it loads, the already existing database table phasedecide is accessed and the value stored in the column installation. If its 0, then we need to go through the installation phase. This form has a :
    a) label1: This label displays the name of the software “Restaurant Management System”.
    b) pictureBox1: This picture box displays the image.
    c) timer1: This timer actually determines the time (300 ticks) for which this splash screen will be visible.
2) **Form1**: After the splash screen, in case installation phase is selected, the form1 is shown. When this form loads, database tables: users and restname in a schema restman are created. Here the user can enter the name of the restaurant (mandatory) and its address in allotted text boxes and they will be stored in the restname table on clicking the button labeled Next.

In case, the button labeled Quit is clicked the installation process is interrupted and ‘installation’ value in phasedecide is maintained at 0.

3) **Form2**: When the super user clicked the button Next on Form1 and name of restaurant is non-empty, this form is shown. Here the super user is provided text fields where he enters his name, username, password, confirm password (so that only intended string is set as password), security question (initialized to “What is your favourite colour?”) and its answer for password retrieval. Of these, none can be empty. On form loading, progress bar is set to 5. On clicking the button labeled Quit (in any of the installation forms) all the tables dynamically created till then are dropped and and ‘installation’ value in phasedecide is maintained at 0. On clicking Next button, the entries of the text boxes are stored in the table users (if all fields are non-empty and the password matches with confirm password field).This form is hidden and form3 shown.

4) **Form3**: This form allows the super user to select the cuisines available in the restaurant. When the form loads, the progress bar value sets at 20. The user can

select any of the predefined cuisines by selecting corresponding check boxes. Select all and Deselect all options are provided for user convenience. In case (s) he wants to add their own cuisine, add more button needs to be clicked which makes the dataGridview1 visible. On clicking Next button, all the selected entries (if any) from the CheckedListbox1 and those from dataGridview1 are concatenated in a string s and stored in the table restname's column cuisines. In case none are selected or entered, General is selected by default which contains a predefined menu of some miscellaneous items. This form is then hidden and form4 shown.

5) **Form4**: Here the super user can add or remove any previously added employees' login information. On loading of this form, progress bar sets to 40.For adding employees, corresponding button is clicked which shows the form5. Similarly form6 is shown to remove employees. Clicking Next button hides the present form and shows form7.
6) **Form5**: This form adds an employee. Entries similar to that for the super user login are stored for every employee to be added into the same table: users

under same constraints. The username and password are case-sensitive. Multiple employees can be added from this form without navigating back and forth between frorm4 and form5. On clicking Add Another Employee, and if the employee is added successfully, same form is shown but with all the text fields cleared and a label notifying the user that previous employee was added successfully. On closing this form form4 is accessed.

7) **Form6**: This form helps the super user to delete any employee added in the previous step. For this, when this form is opened from form4, for all rows of the table users except the first where super user information is stored, the employee name is transferred to cells[0] and employee user name to cell[1] of the dataGridView1. The user can select the checkboxes in the third cell, which will be deleted from the table users when OK is clicked. On closing this form, form4 will accessed again. (Form4 can be accessed in between also).
8) **Form7**: In this form, some generic details about the restaurant are fixed. When the form is loaded, progress bar equals 60 and global int variables h, d, g are initialized to 0. Whenever any of the checkboxes

corresponding to the order types is selected, corresponding variables are set to 1. When the button Next is clicked, it is checked whether the no. of tables and the max. cancellation time are valid numbers, extra charge (if any) for order type is in Rupees or percentage of bill. If these are satisfied, then this information is stored in the table: restname in corresponding columns as strings. If none of the order types is selected, Dine-in is selected by default. Then this form is hidden and form8 is shown.

9) **Form8**: This form fixes the menu. When the form loads,progress bar is set to 70. If the user wants to manage the menu, cuisines are extracted from the restname table (delimiter ';') and put in the combo box. When the user selects a cuisine, its table, if not already existing is created, and all its items displayed in the dataGridView. Then the table is truncated. When the button 'Manage Another Cuisine' is clicked, all those items whose Cells[3] is not selected are then added to the table if all the prices are numbers. On clicking Next, All data till last clicking of the 'Manage Another Cuisine' is already in the database. This form is hidden and form26 is shown.

10) **Form27**: This form fixes the taxes.When the form loads,progress bar is set to 80. VAT is selected by default. Others: Service charge and service tax are optional. If check boxes are clicked, corresponding taxes need to be fixed as a percentage. These are stored as a string in the table : restname. On clicking next, form is hidden and form9 is shown.

11) **Form9**: Then comes the last form of the installation phase where the discounts are decided upon. When this form loads, progress bar is set to 95.User can select any of the offers by selecting the checkbox, and when he does so, corresponding ai (a1 to a8 are global variables that are initialized to zero when the form loads) is set to 1, then fix the discount as a percentage of the sub total of the bill which is stored in the restname table as a ';' delimited string separately for each offer on clicking Finish. Also on clicking Finish button, Installation completes when we create tables:login, customercard and billrecords. The value 'installation' is reset to 1 to indicate: software already installed. This form is hidden and the form10 is shown.

Daily Phase

1)**Form10**: The daily phase of the software begins just after installation or when the ‘installation’ value stored in phasedecide table is found to be 1 when the splash screen loads the form10 is shown. Here any employee can log in. When the username and the password are entered and Log in button clicked, search is executed in the table users for a row with given username and password. If found, login table is updated and useridlogin (global variable) is set to sid value of the logged in employee from the table.. In case forgot password link label is clicked, form11 is shown. On quitting, the application exits (in any following form also). log_out (global variable) is set to 0. For any form from now onwards, we have a menu bar with options to: Log out (if logged in), Quit, go to Home form of the logged in, Instructions and About (form 14 is shown).

2) **Form11**: This is password retrieval form.The user enters his username and then clicks on Get Security Question button to get it from table: users. Then he has to answer correctly and click button2 to get his password (if his username and answer are actually valid). Then he can go back to form10 and try logging in again.

General Employee

1) **Form12**: This form serves as the Home form for the general employees. It is here that the employees log in to and it is here that the employees return to after completion of an operation to perform any further possible operations. When this form loads, the customer count gets stored in the global variable billno and the timings are stored in variable 'to' and 'from' from the tables: phasedecide and restname respectively. Whenever the value of log_out becomes 1, this symbolizes that a form reached from this form has closed because user logged out, so this form also closes. On clicking the button labeled Generate Bill, this form is hidden and the form13 is shown. When the button for placing orders is clicked those for Dine-in, Take-out and Home-Delivery become visible. Similarly when the Change Order button is clicked, buttons for Add and Delete Items become visible. A user defined function timevalidity() checks whether the present time is inside the Restaurant open timings fixed in settings. If so, it returns a 1, otherwise 0.

When the button for Customer Card is clicked, form 15 is shown and this form is hidden. To delete item form16 is shown . to add an item, when the button is clicked timevalidity() function is called and if it returns a 1, the form23 is shown, add_item is set to 1. When any of the buttons for take-out or home-delivery is clicked on, its checked whether timevalidity() function returns a 1 and if it does go to form23 and set table to 2 or 3 respectively. However, when Dine-in is clicked further we need to check if a variable asdf is 0, because this means that no. of tables available is non-zero which in turn shows form23 with tables=1.

2) **Form13**: This is the form where the information of the transaction for which the bill is to be generated is gathered. The transaction no. is entered by the user and OK button is clicked. If it's found to be valid all the check boxes for the offers available are enabled. When the check box corresponding to Customer card is selected, the customer card no can be entered which when deemed to be valid by button3 lets clicking of birthday and frequency offers. In the end, a ';' delimited string is generated

to denote the offers applicable. Clicking of button1 will do this and take to form27.

3) **Form14**: it works as the about box specifying information about the software.
4) **Form15**: In this form information of a customer is stored in the table: customercard. It makes sure that the contact information is a number.
5) **Form16**:This form deletes an item from a previous order. When the form loads, the 'cancellation' value is extracted from the restname table. It stores the maximum order cancellation duration. At this point, only the transaction no. can be entered. All the other controls are disabled. In case it is found to be valid as well as within acceptable cancellation duration, its order is extracted from the table: billrecords, Presented in the dataGridView. Those items that need to be deleted by the customer can be selected by selecting the checkbox and on clicking the Delete items button, billrecords table is updated.
6) **Form23**: This is the form where order is placed or dish items are added. When the form is loaded, if add_items =1 , this means that the form was accessed after clicking Add Items. So the rest of the controls are disabled till it is certain that the

transaction number added is a valid one. On the other hand, add_items=0 means that we are placing an order. So transaction number is generated and the controls are all enabled. The datagridview contains the menu of the cuisine selected in the combobox. Order is taken as a ';' delimited string which gets updated in the table billrecords on clicking the bottom button. This then takes us back to the Home form. On quitting here, the database table: login shows the employee as logged out at this moment.

7) **Form27(1)**: In this form, the information about the transaction is extracted from various tables of the database, and put in appropriate labels. The order is displayed in the DataGridView. Elaborate taxes, discounts and grand Total is calculated. On clicking the generate textfile option, textfile of the name of transaction_no.txt is created in the folder "BillsGenerated" in the same directory as the software .exe file. This file can be printed using the second button.
   I. Service charge (if fixed and applicable) is calculated on the sub total (after deducting discount).

II. Service Tax (if fixed and applicable) is calculated on the service charge paid by the customer.
III. VAT is calculated on the sub total (after deducting discount) and service charge separately for food items and alcoholic beverages (should have type: alcohol)
IV. Grand total is calculated by adding above three to the sub total and deducting the discounts

Limitations

The present version of Restaurant Management System has a few limitations which can be rectified in the future versions.

1) In form8, the user can alter the menu. While doing this he can just empty the contents of a cell of the dataGridView. This will cause an error when the button ‘Manage Another Cuisine’ is clicked and the unselected contents of the dataGridView are added to the table. So it all depends on the user to not to do that.
2) There is no option provided to change one’s password.
3) The software does not segregate the food items into: breakfast, lunch and dinner. According to this software, all the food items are available anytime when the restaurant is open. The employee or the super user has to take care of this if some item is not available at some time.
4) In form15 it is not checked whether the number entered in the textBox2 i.e. corresponding to the contact number is valid or not. Eg: If he gives 7

digits(but u can’t give more than 10 digits and it has to be a number →this is checked)

5) Super User settings not implemented. i.e. the option to show customer, employee, billing records is not implemented.
6) The menu bar item ‘Instructions’ needs to be disabled as it could not be implemented. The user can manually open the user documentation which in pdf format and this will be given with the software.
7) The GUI needed to have images as in pictures shown in User Manual but could not be implemented in the final product due to some resource file errors.
8) The home page opens twice on logging in . So need to log out twice.

Global Variables

1) Those available in all forms.

   int add_item=0;

   int log_out=0;

   int billno=1;

   int tables=0;

   int useridlogin=0;

   And for all the ‘offers’ there are 8 more global variables.

2) Among others, there are those which are global for all the functions of a form. The main ones are mentioned in this document in blue colour.