System Design Document

Catering Reservation System

for
Martha's 'Mazin Catering

Prepared 12/1/2011

by
Bruce Norman

Table of Contents

Introduction

Martha's 'Mazin Catering currently uses an Excel spreadsheet to collect and organize the information about her business and its' clients, events, and recipes. As her business has grown, this method has become inefficient due to the number of records she needs to search and maintain, and is now inadequate to manage her company's information needs.

This project will create a new catering reservation system using a Microsoft Access database and front end user interface to increase productivity and reduce the amount of time currently required to meet and maintain the companies' information requirement needs.

Inputs

- **frmCATEGORY**

  frmCATEGORY is used to input new categories for recipes, such as Appetizers, Breads, Entrees, Soups, etc. Categories are generally an overall description or classification of the recipes that they are intended to encompass. The only input for this form is the new category name to be defined.

- **frmCUSTOMER**

  frmCUSTOMER is used to input or revise pertinent information about a new customer, including their name, address, contact information, and any special notes regarding their preferences, etc. This form has a subform to identify any previous or currently scheduled events for each customer.

  This form also has a search function that will bring up the first record it matches. If there is no match, it defaults to the first customer in the database.

- **frmEMPLOYEE**

  frmEMPLOYEE is used to insert or revise data about the employees of Martha's 'Mazin Catering. These fields are currently limited to names, job title, and salary.

  This form also has a search function that will bring up the first record it matches. If there is no match, it defaults to the first employee in the database.

- **frmEVENT**

  frmEVENT is the used to create, revise, or delete events for customers. This form allows the customer, theme, and location of the event to be chosen from existing lists. New values need to be entered on their respective forms prior to being selected on this form. Additional information about the event, such as date, time, number of guests, date of initial customer contact, and price quote can also be entered on this page.

  This form includes a tabbed subform titled Employees and Recipes, which identify the employees assigned to cater the event and recipes approved by the customer for the event.

- **frmEVENT_EMPLOYEE**

  frmEVENT_EMPLOYEE is a simple form used to identify the employee at a specific event. This form can be used to assign employees to an event.

- **frmEVENT_RECIPE/MENU**
  frmEVENT_RECIPE/MENU is used to identify the recipes in the menu used at a particular event. This form can be used to establish a menu for an event.

- **frmLOCATION**

  frmLOCATION is used to input information about event locations, including their names, what type of location they are (i.e. Business, Church, School, etc.) and any special rules that must be observed at each location.

  This form contains a subform named Catered Events that identifies events that have occurred or are schedule to occur at the current location.

  This form also has a search function that will bring up the first record it matches. If there is no match, it defaults to the first location in the database.

- **frmRECIPE**

  frmRECIPE is used to add, revise, or delete recipes in the catering database. The form allows the selection of a recipe category, which is followed by the name of the recipe and how much it costs to prepare per serving.

  This form also has a search function that will bring up the first record it matches. If there is no match, it defaults to the first recipe in the database.

- **frmTHEME**

frmTHEME is used to add new themes to the database. The only field that is used on this form allows for the input of a theme description, such as anniversary, birthday, holiday, etc.

Outputs

- **rptCUSTOMER_MAILING_LABELS**

  This report creates a complete record set of mailing labels that includes all the customer names and addresses in the database.

- **rptEVENT_MENU_DETAILS**

  This report creates a detailed list of all the events in the database, including the customer, customer contact information, location, theme, time and date, number of guests, price quote, and a complete menu of the items to be served at the event. The menu list includes the category and name of the recipe, number of servings, price per serving, and a total price of each individual item. These individual totals are calculated at the end of the list to account for all menu costs.

- **rptEventbyCustomer**

  This report creates a single transaction record of events related to a customer using a parameter query. The report asks for a customer name, which then returns a list of each event, the date of the event, its location, and the customer name, address, and contact information.

Procedures (Internal)

Macros

- **frmSplashScreen**
  On Timer

- **frmCATEGORY, frmCUSTOMER, frmEMPLOYEE, frmEVENT, frmEVENT_EMPLOYEE, frmEVENT_RECIPE/MENU, frmLOCATION, frmRECIPE, frmTHEME: Navigation Buttons**
  First Record
  Last Record
  New Record
  Save Record
  Delete Record
  Close Form

Event Procedures

- **Next Record** – used on frmCATEGORY, frmCUSTOMER, frmEMPLOYEE, frmEVENT, frmEVENT_EMPLOYEE, frmEVENT_RECIPE/MENU, frmLOCATION, frmRECIPE, frmTHEME

```
Private Sub cmdNextRecord_Click()

On Error GoTo Err_Next_Record_Click
   'go to next record
   DoCmd.GoToRecord , , acNext

Exit_Next_Record_Click:
   Exit Sub

Err_Next_Record_Click:
   MsgBox Err.Description
   Resume Exit_Next_Record_Click

End Sub
```

- **Previous Record** – used on frmCATEGORY, frmCUSTOMER, frmEMPLOYEE, frmEVENT, frmEVENT_EMPLOYEE, frmEVENT_RECIPE/MENU, frmLOCATION, frmRECIPE, frmTHEME

```
Private Sub cmdPreviousRecord_Click()

On Error GoTo Err_PreviousRecord_Click
```

```
    'go to previous record
    DoCmd.GoToRecord , , acPrevious

Exit_PreviousRecord_Click:
    Exit Sub

Err_PreviousRecord_Click:
    MsgBox Err.Description
    Resume Exit_PreviousRecord_Click

End Sub
```

- **Exit** – used on frmCATEGORY, frmCUSTOMER, frmEMPLOYEE, frmEVENT, frmEVENT_EMPLOYEE, frmEVENT_RECIPE/MENU, frmLOCATION, frmRECIPE, frmTHEME

```
Private Sub cmdExit_Click()
Result = MsgBox("You are about to exit Access." _
            & vbCrLf & "Do you wish to continue?" _
            , vbQuestion + vbYesNo, "WARNING!")
If Result = vbYes Then
    DoCmd.Quit
End If
End Sub
```

Functions

- **CapAll**
  This function is used on the frmCUSTOMER to capitalize the letters of any values input into the Customer Name and State fields. CapAll is defined in the basCapitalize module using the UCase built in function.

```
Private Sub txtCustomerName_AfterUpdate()

'Capitalize the Cust_Name field value
    If Not IsNull([txtCustomerName]) Then
        [txtCustomerName] = CapAll([txtCustomerName])
    End If

End Sub
```

```
Private Sub txtCustomerState_AfterUpdate()

'Capitalize the Cust_State field value
   If Not IsNull([txtCustomerState]) Then
      [txtCustomerState] = CapAll([txtCustomerState])
   End If

End Sub
```

- **Find**
  This function is attached to the On Click event of the btnFind ("Find Customer", etc.) on several forms, including frmCUSTOMER, frmEMPLOYEE, frmLOCATION, and frmRECIPE. This function will bring up the first record similar to the input value.

```
Private Sub btnFind_Click()

   Set rs = Me.Recordset.Clone
   rs.FindFirst "[Cust_Name] Like '*" & Me![txtFind] & "*'"
   Me.Bookmark = rs.Bookmark

End Sub
```

- **Repeat Customer**
  This function is used on the On Current event of the frmCUSTOMER. This function will identify if the current record is a repeat customer by searching the customer notes for those words. If it finds Repeat Customer in the notes, a red label box is made visible to the right of the Customer Name with the words Repeat Customer in white. (Ideally I wanted to do this by counting the number of records in the subform on the page, but I never got this code to work.)

```
Private Sub Form_Current()

   'For displaying if customer is a repeat customer
   If [Cust_Notes] Like "*Repeat Customer*" Then
      lblRepeatCustomerMsg.Visible = True
   Else
      lblRepeatCustomerMsg.Visible = False
   End If

End Sub
```

Queries

- **qryCUSTOMER_EVENTS**

  This query has the following fields from tblCUSTOMER, tblEVENT, and tblLOCATION:

  Customer_ID, Event_ID, Event_Date, Event_Time, Loc_Description

  It is used for subCUSTOMER_EVENTS in frmEVENTS.

- **qryEVENT_CUSTOMER_DATE_BY_PARAMETER**

  This query has the following fields from tblCUSTOMER, tblEVENT, tblLOCATION:

  Event_ID, Cust_Name, Event_Date, Loc_Description, Cust_Phone, Cust_Email, CustFContact, Cust_LContact.
  This query adds the additional following fields:

  Address: IIf(Nz([Cust_Address2],"")="",[Cust_Address1],[Cust_Address1] & Chr$(13) & Chr$(10) & [Cust_Address2]), CityStateZip, and WholeAddress using
  It is used for rptEventbyCustomer

  CityStateZip: IIf([Cust_City],[Cust_City] & ", ","") & [Cust_State] & " " & [Cust_Zip]

  WholeAddress: [Address] & Chr$(13) & Chr$(10) & [CityStateZip]

  It is used for rptEVENTbyCUSTOMER.

- **qryEVENT_EMPLOYEE_NAME**

  This query has the following fields from tblEVENT and tblEMPLOYEE:

  Event_ID, Employee_ID, Emp_FName, Emp_LName, Employee Job Title

  It is used to create subEVENT_EMPLOYEE_NAME on frmEVENT.

- **qryEVENT_MENU_DETAILS**

This query has the following fields from tblCATEGORY, tblRECIPE, tblEVENT_RECIPE/MENU, tblEVENT, tblTHEME, tblCUSTOMER, and tblLOCATION:

Event_ID, Cust_Name, Cust_Phone, Cust_Email, Cust_FContact, Cust_LContact, Event_Date, Event_Time, Event_Guests, Event_Price_Quote, Loc_Description, Category_Name, Recipe_Name, Number_Servings, Price_Per_Serving

This query adds the additional following fields:

Item Total: [Number_Servings]*[Price_Per_Serving]

It is used for rptEVENT_MENU_DETAILS.

- **qryLOCATION_EVENTS**

  This query has the following fields from tblTHEME, tblLOCATION, tblEVENT, tblCUSTOMER:

  Event_ID, Location_ID, Event_Date, Cust_Name, Theme_Description

  It is used for subLOCATION_EVENTS on frmLOCATION

- **qryRECIPE_ID_CATEGORY_NAME_$**

  This query has the following fields from tblCATEGORY, tblRECIPE, tblEVENT_RECIPE/MENU:

  Event_ID, Recipe_ID, Category_Name, Recipe_Name, Price_Per_Serving

  It is used to create subRECIPE_ID_CATEGORY_NAME_$ on frmEVENT.

Procedures (External)

General Flow of the Activities

The current process of booking a new catering event begins with the initial customer contact. This is handled by the event coordinator, who begins documenting the event by searching for the customer in the existing spreadsheet. This is followed by defining the event theme, location, and other necessary information such as date, time, location, theme, menu, number of guests, etc. The event coordinator determines which employees will work a specific event and is responsible for reserving the location venue.

The owner and the food coordinator design the menu from the recipes in the existing spreadsheets based on the event theme. This menu is then approved by the customer and used to determine what supplies will need to be ordered for each recipe and event. Each recipe is sorted by category, such as entrée, vegetarian, etc., and the price for each menu item is calculated based upon the number of guests. This information is included on the event detail form. The food coordinator is responsible for managing the kitchen staff to ensure each menu item is properly prepared for each catering event.

The financial aspects of the company are maintained by an accountant and bookkeeper, while the payroll is managed by an independent, outside firm. Only those details that deal specifically with catering functions will be incorporated into the new system. Recipes and their ingredients are maintained in a separate filing system and will not be included in the new catering reservation system.

The Usage of the System

The new catering reservation system needs to efficiently document the related information for each catered event, including customer, location, theme, menu, and assigned employees. It should be capable of calculating the total of each menu item based upon number of guests and allow the user to efficiently locate requested information for each event. The new system should allow cross-referencing certain requested items, such as being able to view previous events at a particular venue or view recipes for a specific theme, to assist in planning for the new event. It should minimize data entry errors by using appropriate input masks and formats and must be capable of producing reports such as event details for specific customers/events.

The new system will be a simple data entry and retrieval system, thus new users or those with little or no exposure to Microsoft Access will require only minimal training to become proficient in its operation. The new system will save significant amounts of time

currently associated with planning new catering events and retrieving or cross-referencing requested information.

Usage of the new system has been simplified by using a main menu navigation form, which will appear after a welcome splash page with a four second timer. The navigation form uses two levels of horizontal tabs at the top of the page. The uppermost tabs include the Forms, Queries, and Reports categories. By clicking on each of these, the individual documents contained within each category will be displayed in the secondary tabs below them. Clicking on these secondary tabs will display the related document on the screen below them. For example, by clicking on the Forms tab in the top row and then the Customers tab in the secondary row, the Customer form will be displayed on the screen.

Once a form is selected and displayed, the user can navigate through existing records by using the upper navigation buttons (First Record, Previous Record, Next Record, and Last Record). The user can use the navigation buttons to the left of the displayed form to Add Record, Save Record, Delete Record, or Close Form. Closing the form will reopen the navigation form with its default form selected. Selecting the Exit button on any of the forms will close Access.

Selecting the Queries tab will open a form that has a list box with all existing queries available. Users can select a query and then chose to either Preview it (which creates a preview of how the document will appear if printed) or Display it (which brings up an editable version of the query).

The Reports tab allows the user to view pre-defined documents that confirm or request information from the database. These reports can also be generated based upon a response to a request for a specific parameter, such as a customer name to determine which events belong to them.

Interface Design and Coding Standards

External

- **Fonts:** Minimum size of 11 and the maximum size of 20 in Arial will be used consistently across all forms (except for logo, which uses Edwardian Script). Larger text will be used for document and tab titles. Labels (size 11), tabs(size 14), and command buttons(size 11) will have bold font, text boxes will have normal font

weight. The title of each form will be bold font (size 20). Fonts will use custom colors as denoted below to contrast their individual background colors.

- **Colors:** Custom colors will be used for backgrounds, fills, fonts, command buttons, labels, text boxes, list boxes, combo boxes, and menus. Custom colors to be used are:
    - **Dark Slate Gray** #2E2633, R046 G038 B051
    - **Light Golden Yellow** #EFFFCD, R239 G255 B205

  Usage of colors for font will depend upon the background colors and will use the opposite custom color for clarity.

  The "Find" function buttons will be red with contrasting white text.
- **Graphics:** The logo of Martha's 'Mazin Catering will be included in all of the main windows that are not dialogue or messages boxes.
- **Size of Forms and Buttons:** Forms will be a minimum of 7" wide by 5" tall. The background color of forms and screens will use custom colors as noted above. The goal will be to have the text contrast with the background in order to create a readable screen without causing visual discomfort to the reader. Custom navigation buttons will conform to standard heights (.5") and widths (1.15"), while the "Find" function button will be slimmer for intended contrast.
- **Command Buttons:** Command buttons will be used for navigation macros and event procedures, as well as for "Find" function buttons. Text will be used to identify the functions of various command buttons.
- **List Boxes:** List boxes will be used to select data from a set of predetermined choices to be entered into a system as a part of a record.
- **Drop Down Lists:** Drop down choice lists will be used for defined choices the user must make.
- **Tool Tips:** Tool tips will be added to form elements as an aid to user navigation. The approach will be making sure the user has adequate information and guidance without adding clutter to the interface. Additional design elements and styles may be added to these standards as necessary.
- **Properties:** The following properties of each form will be checked as denoted:
    - **Auto Center: "**Yes**"**

- **Auto Resize: "**Yes**"**
- **Fit to Screen: "**Yes**"**
- **Record Selectors: "**No**"**
- **Navigation Buttons: "**No**"**
- **Scroll Bars: "**Neither**"**
- **Control Box: "**No**"**
- **Close Button: "**No**"**
- **Min Max Buttons:** "None"

- **Labeling:** Labels for text boxes will be aligned in a stacked arrangement, placing the label for each text box to their immediate left. The only exception for this is the txtFind text box for the "Find" function, which is located to the left of its corresponding button. This is meant to imply an order of operations, i.e. the text box must have a value before clicking on the button.

Internal

Program Documentation

A descriptive comment will be included for every procedure to explain its' overall purpose. A short comment may be included to clarify lines of complex code.

Naming Convention for Code Variables and Objects

Hungarian naming convention will be used to name variables and objects as recommended by Microsoft, except as noted below. The names will contain no spaces, and will start with a lower case letter and include an upper case letter for the first letter of a new distinguished word; for example "lblFirstName", or "txtReportDate".

- "rpt" for reports
- "qry" for queries
- "frm" for forms
- "sub" for subforms
- "tbl" for tables
- "bas" for modules

- "txt" for text
- "lbl" for labels
- "lst" for list boxes
- ""cbo" for combo boxes
- "cmd" for command boxes
- "img" for images
- "lin" for lines
- "shp" for rectangles
- **Exceptions:** Tables, reports, queries, forms, and subforms will use qryCUSTOMER_EVENTS (as an example) as the way they are named, with the exception of the two "navigation" forms, frmNavigation and frmSplashPage. These two forms will use the same Hungarian naming convention as other database objects to distinguish them from the others.
- Attributes in the tables will use Category_ID (as an example) as the way they are named.

Naming Conventions for Database Objects

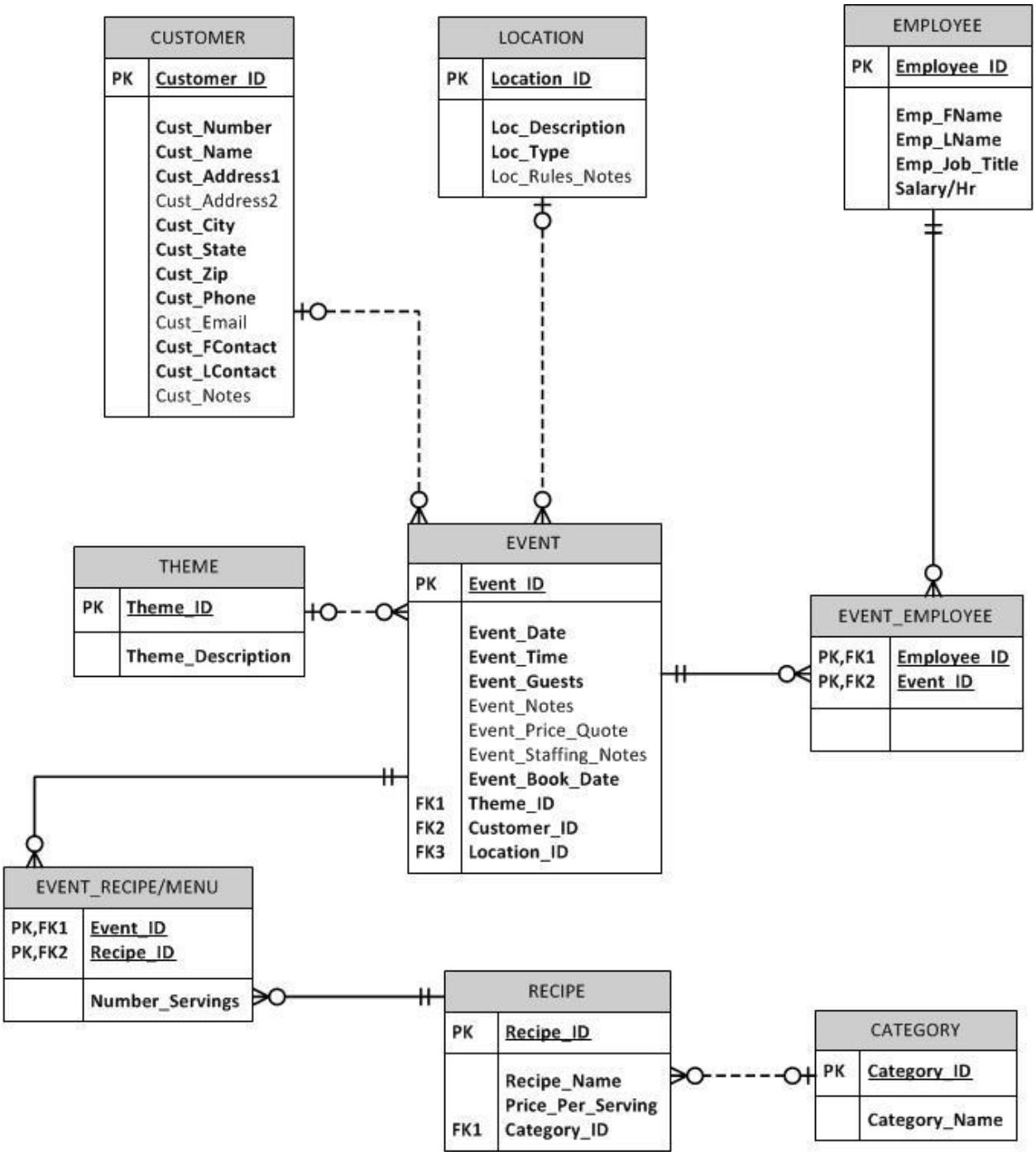
The database will use the same naming conventions for code objects and variables.

- "rpt" for reports
- "qry" for queries
- "frm" for forms
- "tbl" for tables
- "mcr" for macros
- "bas" for modules

All table primary keys will use a unique numbering system for identifying the primary key.

APPENDICES

Appendix A - Entity Relationship Diagram (ERD)



Appendix B - Metadata Dictionary

Meta Data Dictionary

ENTITY	ATTRIBUTE	DEFINITION	DOMAIN	REFERENTIAL INTEGRITY
CATEGORY	Category_ID	Unique Identifier for each recipe category	System assigned. Unique. Numeric (10)	PK(Primary Key)
	Category_Name	The category of a recipe	Required. Non-Unique. Char(20). Lookup: Valid values = Appetizer, Veverage, Bread, Dessert, Holiday, Entrée, Salad, Seasonal, Side Dish, Soup, Vegetarian, Other.	

ENTITY	ATTRIBUTE	DEFINITION	DOMAIN	REFERENTIAL INTEGRITY
CUSTOMER	CustomerID	Unique Identifier for each Customer	System Assigned. Unique. Numeric(10)	PK(Primary Key)
	Cust_Number	Customer number generated by company	Required. Non-Unique. Char(10).	
	Cust_Name	Name of company, individual or organization ordering event	Required. Non-Unique. Char(50).	
	Cust_Address1	Customer address line one	Required. Non-Unique. Char(40).	
	Cust_Address2	Customer address line two	Optional. Non-Unique. Char(25)	
	Cust_City	Customer city	Required. Non-Unique. Char(25).	
	Cust_State	Customer state	Required. Non-Unique. Char(2).	
	Cust_Zip	Customer zip	Required. Non-Unique. Char(5).	
	Cust_Phone	Customer contact phone	Required. Non-Unique. Input Mask: Phone (###)###-####	
	Cust_Email	Customer contact e-mail	Optional. Non-Unique. Char(40).	
	Cust_FContact	First name of customer contact	Required. Non-Unique. Char(15).	
	Cust_LContact	Last name of customer contact	Required. Non-Unique. Char(20).	
	Cust_Notes	Notes regarding customer	Optional. Non-Unique. Char(255)	

ENTITY	ATTRIBUTE	DEFINITION	DOMAIN	REFERENTIAL INTEGRITY
EMPLOYEE	EmployeeID	Unique Identifier for each Employee	System assigned. Unique. Numeric (10)	PK(Primary Key)
	Emp_FName	Employee first name	Required. Non-Unique. Char(20)	
	Emp_LName	Employee last name	Required. Non-Unique. Char(20)	
	Emp_Job_Title	Employee position/job title	Required. Non-Unique. Char(15) Lookup: Valid Values = Chef, Dishwasher, Wait Staff, etc	
	Salary_Hr	Employee salary per hour	Required. Non-Unique. Char(6)	

ENTITY	ATTRIBUTE	DEFINITION	DOMAIN	REFERENTIAL INTEGRITY
EVENT	EventID	Unique Identifier for each Event	System assigned. Unique. Numeric (10)	PK(Primary Key)
	LocationID	Identifier for Event's Location	Required. Non-Unique. FK(LOCATION)	An EVENT cannot exist without a related LOCATION record
	CustomerID	Identifier for Event's Customer	Required. Non-Unique. FK(CUSTOMER)	An EVENT cannot exist without a related CUSTOMER record
	ThemeID	Identifier for the event's Theme	Required. Non-Unique. FK(THEME)	An EVENT cannot exist without a related THEME record
	Event_Date	Date event will be held	Required. Non-Unique. Input Mask: Date mmmm/dd/yyyy	
	Event_Time	Time event will be held	Required. Non-Unique. Input Mask: Long Time	
	Event_Guests	Number of guest attending event	Required. Non-Unique. Numeric(4)	
	Event_Notes	Notes on the event	Optional. Non-Unique. Char(255)	
	Event_Price_ Quote	Initial quote on the event cost	Optional. Non-Unique. Numeric(6)	
	Event_Staffing_Notes	Notes on staff needed for event	Optional. Non-Unique. Char(255)	
	Event_Book_ Date	Date customer first contacts company to book event	Required. Non-Unique. Input Mask: Date mmmm/dd/yyyy	

ENTITY	ATTRIBUTE	DEFINITION	DOMAIN	REFERENTIAL INTEGRITY
EVENT_ EMPLOYEE	EmployeeID	Identifier for Employee working at each event	Required. Non-Unique. FK(EMPLOYEE)	CPK (Composite Primary key) An EVENT_EMPLOYEE cannot exist without a related EMPLOYEE record
	EventID	Identifier for each Event	Required. Non-Unique. FK(EVENT)	CPK (Composite Primary key) An EVENT_EMPLOYEE cannot exist without a related EVENT record

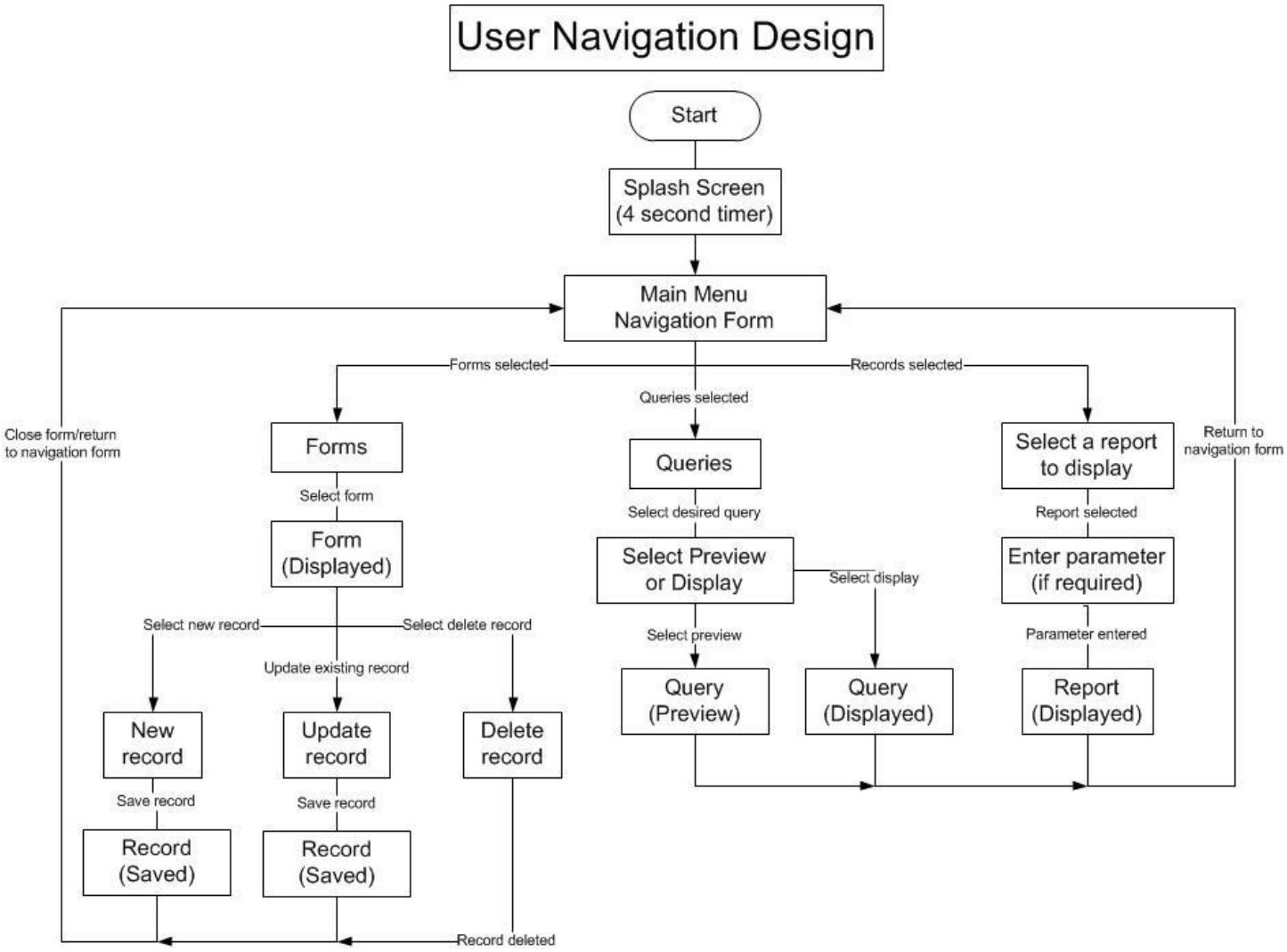
ENTITY	ATTRIBUTE	DEFINITION	DOMAIN	REFERENTIAL INTEGRITY
EVENT_ RECIPE	EventID	Identifier for each Event	Required. Non-Unique. FK(EVENT)	CPK (Composite Primary key) An EVENT_RECIPE cannot exist without a related EVENT record
	RecipeID	Identifier for each Recipe	Required. Non-Unique. FK(RECIPE)	CPK (Composite Primary key) An EVENT_RECIPE cannot exist without a related RECIPE record
	Number_Servings	Number of servings ordered for each recipe	Required. Non-Unique. Numeric(4)	

ENTITY	ATTRIBUTE	DEFINITION	DOMAIN	REFERENTIAL INTEGRITY
LOCATION	LocationID	Unique identifier for each location	System Assigned. Unique. Numeric(10)	PK(Primary Key)
	Loc_Description	Name or description of location	Required. Non-Unique. Char(50)	
	Loc_Type	Location Type	Required. Non-Unique. Lookup: Valid Values = Hall, Church, Private Residence, etc	
	Loc_Rules_Notes	Rules associated with location	Optional. Non-Unique. Char(150)	

ENTITY	ATTRIBUTE	DEFINITION	DOMAIN	REFERENTIAL INTEGRITY
RECIPE	RecipeID	Unique identifier for each Recipe	System Assigned. Unique. Numeric(10)	PK(Primary Key)
	CategoryID	Identifier for each recipe's Category	Required. Non-Unique. FK(CATEGORY)	A RECIPE cannot exist without a related CATEGORY record
	Recipe_Name	Name of recipe	Required. Non-Unique. Char(50)	
	Recipe_ PricePerServing	Cost of recipe per serving	Required. Non-Unique. Numeric(6)	

ENTITY	ATTRIBUTE	DEFINITION	DOMAIN	REFERENTIAL INTEGRITY
THEME	ThemeID	Unique identifier for each event's Theme	System Assigned. Unique. Numeric(10)	PK(Primary Key)
	Theme_Description	Description of each theme	Required. Non-Unique. Lookup: Valid Values = Wedding, Birthday, Anniversary, etc	

Appendix C - Navigation Design



Appendix D – Provided Samples

Martha's 'Mazin Catering

Customer Number: 1989
Date Taken: 4/19/10

Event Detail

Event Date:	June 18, 2010	Price Quoted:	$7,250.00
Event Location:	Edmonds Community College - Tub 204	Total Employees Needed:	Wait staff to keep buffet filled
Event Time:	9:00 PM (After Graduation Ceremony)	Employee:	Marion Dennis – Wait Staff
Event Theme:	Graduation		Andrea Collins – Wait Staff
Number of Guests:	100		David Beck - Chef
Event Notes:	This event is for instructors and administrators of the college after the graduation ceremony. Individuals will arrive from 8:30 to 9:00 PM To be buffet style serving. Hors d'œuvre Main course Light Dessert		_____ _____ (list additional employees on reverse side)

Customer Detail

Customer Name:	EdCC Teacher's Association	Customer Notes:	Customer requires vegetarian items
Customer Address:	20000 68th Ave W. Snohomish Hall 202 Lynnwood, WA 98036	Contact:	Marti Baker or Peter Farrar
Customer Phone:	425 640 1500		
Customer email:	EdCCTA@edcc.edu		

Menu

Estimated cost: $3,575

Item:	Caramelized Onion Dip with Potato Thins - 200 Vegetarian	Cost:	$400
Item:	Roasted Pepper And Mozzarella With Pesto - 200 Vegetarian	Cost:	$400
Item:	Party Pasta with Buffalo Strips – 200	Cost:	$500
Item:	Vegetarian Lasagna – 100	Cost:	$700
Item:	Hearty Meat Lasagna – 100	Cost:	$825
Item:	Caeser Salad – 200	Cost:	$300
Item:	Assorted Cookies, Brownies – 275	Cost:	$450

Item (list additional menu items on reverse side)