

Please note the changes in blue and the additional 3<sup>rd</sup> page.

### Project requirement and description:

You required to create a collaborative online catering system (COCS) that enables customers/group to place a catering order for their events from multiple registered caterers.

Only registered customers are able to place a catering order, but any customer can view the menu items for each vender, special offers, and list of caterers.

The system has few common pages/features and three main modules, Admin, vender, and customer modules.

The COCS is developed using ASP.NET and SQL database.

#### Common pages: (30 marks)

1. **(5 marks) Homepage:** The home page should display the food items/menu and caterers list. Customers should be able to search, login, or register from the home page.
2. **(2.5 marks) Login:** allows registered customers to login and access any of the customer's module pages.
3. **(5 marks) Register:** allows customers/**caterers** to register to be able to place/**receive** orders
4. **(2.5 marks) Forgot password:** allows the customers to recover their password
5. **(2.5 marks) Reset Password:** allows users to change their password
6. **(7.5 marks) Search:** the search is available from the any page (hint: make it part of a master page) and to anyone using the site as follow:
  - **Admin:** when admin perform a search, s/he can search for any order in the system, Caterers, or food items.
  - **Registered customer/Caterers:** when a registered user performs a search, s/he can search for any of their own orders, or food items.
  - **Non-Registered customer:** when a non registered customer performs a search, s/he can search only for food items.
7. **(5 marks) Customer/Caterer Profile:** customers can view and change their full information they provided at registration, including profile image.  
Note: username and email cannot be updated.

#### Customer module: (20 marks)

1. **(5 marks) My Team:** Customers can share their menu with a collaborative team (friends) to help select the full menu (friends can be sent an invitation link to join the team once they register, or request to join the organization team) either way, friends have the option to accept or decline to request.
2. **(5 marks) Catering packages/menu:** Customers can view the list of caterers and their pre-created catering packages in addition to other menu items, and add packages or menu items to a shopping cart.
3. **(7 marks) Cart:** customers can add, remove, or update items in their shopping carts, all team members have full access to the same shopping cart, to add, delete, or

update any item. Orders are submitted only by the team lead (the creator of the team by default, but it can be transferred to another member in the team)

4. **(3 marks) My orders:** customers can view all their order with a status for each order (Pending, approved, rejected, canceled, being-prepared (only on the fulfillment date), fulfilled)

Note: customers can cancel an approved or pending orders, but not a being-prepared or fulfilled orders)

#### **Caterer module: (20 Marks)**

1. **(5 marks) Orders:** In this section ~~admin~~ the caterer can view the food order details and they have also right to change order status according to current status.
2. **(15 marks) Food packages/Menu:** Create, **update, delete** food packages (i.e. an appetizer, salad, main course, desserts for 20 persons is for \$\$\$) and/or food menu items.

#### **Admin module: (10 Marks)**

1. **(5 marks) Dashboard:** In this section, admin can see all detail in brief like a total order, not confirmed order, confirmed order, total food being prepared, total cancelled orders, and total registered customers.
2. **(2 marks) Reports:** shows reports of orders counts and monthly sales.
3. **(3 marks) Manage Caterers:** All administrators to create (add) a new caterer or **approve/decline a caterer** (if your team choose to allow caterers to register themselves)

#### **General Look, design, detailed features and functionalities that are not clearly listed above (20 marks)**

**See next page for detailed marking schema for the features and functionalities that the project must contain within the above modules or other additional pages.**

#### **Bonus:**

Create 2-4 themes and allow the user to select their desire theme dynamically.

Group Project  
Collaborative Online Catering System (COCS)  
CP380

In addition to the project description above, the project must adhere to the following features and functionalities to get a full mark:

**NOTE: any missing feature/functionality will result in deducting the points in the brackets. A maximum of 100 points can be deducted.**

- 1) (-20) Built using the ASP.NET web forms.
- 2) (-20) Integration with a database (MS SQL, SQL Express, Azure SQL, or local MDF db file).
- 3) Implement the common pages and the tree main modules mentioned above:
  - a) (-20) User module.
  - b) (-20) Caterer module.
  - c) (-10) Admin module.
- 4) (-8) Implement login, forgot password, reset password, and registration
- 5) (-5) Handled login and registration by using the ASP.NET application services (membership, roles, profile).
- 6) (-5) Implement a minimum of two different roles. (Note: Ideally three roles should be used: admin, caterer, Registered user).
- 7) (-5) Final website must be hosted on AZURE sites, or on the cc-cpserver on the confederation college premises (need to let me know if you would like to host on cc-cpserver (web and SQL) so I can set it up for your project).
- 8) (-5) Implement master page(s) and/or nested master page(s).
- 9) (-5) Implement a navigation control (site map path, tree view, or menu).
- 10) Implement the following data controls:
  - a) (-5) DetailsView and/or formView.
  - b) (-7) DataList, GridView, and/or ListView.
- 11) Implement the following functions within the data control (where needed):
  - a) (-2) Select (display data).
  - b) (-2) Delete.
  - c) (-4) Insert.
  - d) (-4) Update.
- 12) Implement the following validation controls where needed:
  - a) (-2) CompareValidator,
  - b) (-2) CustomValidator,
  - c) (-2) RegularEWxpressionValidator,
  - d) (-2) RequiredFieldValidator, and
  - e) (-2) ValidationSummary.
- 13) (-5) Implement a contact us page with sending email capability (or saving it locally).
- 14) (-5) Implement a "Web Form User Control".
- 15) (-5) Maintain a consistent look between pages.
- 16) (**bonus 2**) Implement a shopping cart with add, update, and remove items capabilities.
- 17) (**bonus 2**) Implement an invitation to team functionality, and invitation code.
- 18) (-5) Implement session and/or cookies handling.
- 19) (bonus 2 marks/extra theme) Implement multiple themes.

Group Project  
Collaborative Online Catering System (COCS)  
CP380

In addition to the project description above, the project must adhere to the following features and functionalities to get a full mark:

NOTE: any missing feature/functionality will result in deducting the points in the brackets. A maximum of 100 points can be deducted.

- (-20) Built using the ASP.NET web forms.
- (-20) Integration with a database (MS SQL, SQL Express, Azure SQL, or local MDF db file).
- Implement the tree main modules mentioned above.
  - (-15) User module
  - (-15) Caterer module
  - (-10) Admin module
- (-8) Implement login, forgot password, reset password, and registration
- (-5) Handled login and registration by using the ASP.NET application services (membership, roles, profile).
- (-5) Implement a minimum of two different roles. (Note: Ideally three roles should be used: admin, caterer, Registered user).
- (-5) Final website must be hosted on AZURE sites, or on the cc-cpserver on the confederation college premises (need to let me know if you would like to host on cc-cpserver (web and SQL) so I can set it up for your project).
- (-5) Implement master page(s) and/or nested master page(s).
- (-5) Implement a navigation control (site map path, tree view, or menu).
- Implement the following data controls:
  - (-5) DetailsView and/or formView.
  - (-7) DataList, GridView, and/or ListView.
- Implement the following functions within the data control (where needed):
  - (-2) Select (display data).
  - (-2) Delete
  - (-4) Insert
  - (-4) Update
- Implement the following validation controls where needed:
  - (-2) CompareValidator,
  - (-2) CustomValidator,
  - (-2) RegularEWxpressionValidator,
  - (-2) RequiredFieldValidator, and
  - (-2) ValidationSummary.
- (-5) Implement a contact us page with sending email capability (or saving it locally).
- (-5) Implement a "Web Form User Control".
- (-5) Implement a shopping cart with add, update, and remove items capabilities.
- (-5) Implement an invitation to team functionality
- (-5) Implement session and/or cookies handling
- (-5) Maintain a consistent look between pages
- (bonus 2 marks/extra theme) Implement multiple themes