**Due Date:** Nov 04, 2018 23:59:59       **Max Points:**90  
  
**Details:**

This is a Collaborative Learning Community (CLC) assignment.

**Objective:**Create a management interface for the commerce page administrator to manage customer roles, permissions, and payment information.

**Activity**: Browse the Internet and review several commerce sites to develop a feel for what customer management looks like. Make inferences from the interaction since you will not have behind-the-scene access. Discuss the customer management functionality and make a list of all elements and functions your customer administrator panel should do. Discuss the interface, specific functions, and necessary backend support, including payment processing and a connection with a financial institution. Discuss how the customer administrator interface will visually fit with the rest of the e-commerce project. Discuss the various types of customers, their permissions, and justification for said permissions levels Discuss how the new functionality might work with existing functionality and do not hesitate to modify prior modules. Discuss any limitations you might impose on the system. Revisit the discussion on how you would verify the integrity of the data in your database and what integrity means in the context of this project.

**Build:**

1. In MySQL, build the necessary tables to store the information required by the customer administrator. Modify the schema, the E-R diagrams, the tables, the keys, and the relationships accordingly. You might need to add new fields to existing tables to support new functionality. You might need to split existing tables as well.
2. In PHP, write a Customer Class and a Payment Processing Class that will enable the management of customer data, their permissions, and payment information. Modify the User Registration Page to accodomodate the new roles and set a default role.
3. In HTML, build the minimally functional form to display lists of customers and GUI elements connected with the PHP backend functionality.
4. Write an additional Customer Action class, which will implement the functionality of generating and presenting information about the customer base, including demographic data and purchase history data. Revisit the entire database organization to enable smooth and efficient queries of customer and product tables.
5. In PHP perform the necessary verification of compliance with constraints you decided upon. Before you code, discuss and decide every detail of the user (i.e., the customer administrator) experience such as: how information will be displayed, what happens after an action is taken, what if a wrong action is taken, recovery from errors, etc.
6. Based on the desired interaction and data captured, build the MySQL tables to store this information and enable the desired functionality. Connect this module to previous modules of your database and ensure cohesiveness. Discuss any revisions that might be necessary to code, tables, schema, or user interface created in previous module(s).
7. In PHP, implement a simulated financial entity, to be connected with customer’s chosen method of payment during registration. Simulated the verification and authentication of the customer’s choice of payment methof (i.e., checking account or credit card).

**Deliverables**:

1. The schema and E-R diagram
2. All necessary SQL tables
3. All necessary PHP classes
4. All necessary HTML
5. All necessary Python code, JavaScript, or other language
6. A fully functional New Product Page

**What to submit**:

1. A document describing the project, a list of all modules, files, and a user guide. In the document, explain how desired functionality, features, and constraints have been implemented in code.
2. Upload all necessary files to the GCU Cloud Hosting Solution.
3. In each file, include a commented header with the following information: Project name and version, Module name and version, Programmer(s) name(s), Date, Short synopsis of the module, and References.
4. Comments within the code, explaining non-obvious sections
5. Revise previous project and code files as needed and document revisions made.
6. In LoudCloud, submit the URL to your project.