This is a Collaborative Learning Community (CLC) assignment.

After the client’s needs are captured in User Stories, requirement development is the next step for documentation. Requirements are deconstructions of the user stories into mostly functional behaviors. The requirements identify “what” the application will do. A User Story is a vague representation of an idea or concept a specific user category would like. The requirements document will expand the User Story into multiple functional behaviors that the software must exhibit. For example, in an e-commerce application, a User Story may be “As a customer, I want a convenient way to shop, pay, and ship an order.” This is too vague to implement. This example User Story can be deconstructed into several individual requirements, including identifying web pages to display, GUIs, order of operation, a shopping cart, the ability to save the shopping cart, make a wish list, and so on. A good requirement is clear, concise, and testable. Refer to “EARS – Easy Approach to Requirements Syntax: The Definitive Guide,” located in the Topic Materials, for more details on requirement writing.

Use the Agile-based tool to assign resources to the User Stories you have created, and develop a sprint milestone assignment. In the assignment, each CLC group member will create requirements for the User Stories they are assigned in the Agile-based tool. Create the requirements document using the “CST-326 Requirements Document” as guidance. Some User Stories may be simple, some complex. During the sprint, your CLC group will manage the assigned User Stories to complete the requirements document. Management of Issues may require the reassignment of existing Issues in the Agile-based tool to a different milestone, the creation of new Issues, and/or reassignment of Issues between members of the CLC group.

As applicable, include the following types of requirements: system requirements, user requirements, tool requirements (compiler, programming languages, third party APIs, etc…), and functional performance. To help organize and provide context for the requirements document, you can include discussion and high-level block diagrams or UML diagrams so that the reader of the requirements document has a basic understanding of what the software will do. The reader will also understand the high-level software architecture (e.g., database and network connectivity).

**This assignment has two parts: A Requirements Document and a Milestone Version Description Document**

**Part 1: The Requirements Document**

﻿Use the provided template, “CST-326 Requirements Document,” to capture the project requirements, functional discussion, and architecture diagrams. The template is a guide. Your CLC group will name, add, and delete sections as appropriate for your software.

**Part 2: The Module Version Description**

Use the provided template, “CST-326 Milestone Version Description Document,” located in the Course Materials, to capture the project status. The CLC Milestones are effectively client deliveries. Many projects require a document that summarizes the current delivery and project status. This document is used as an artifact to identify the work the CLC group did during the sprint. You will use data from Agile-based tools and your Git repository.

Prepare this assignment according to the APA guidelines found in the APA Style Guide, located in the Student Success Center. An abstract is not required.

This assignment uses a rubric. Please review the rubric prior to beginning the assignment to become familiar with the expectations for successful completion.

You are not required to submit this assignment to LopesWrite.

Submit the assignment to LoudCloud as directed by your instructor.