Create a small enterprise application using .NET MVC given the following business requirements:

Create a Bible Verse application.

Create a Bible Verse Entry page with a data entry single form that contains the following form elements:

1. Testament Selection (i.e. Old or New Testament)
2. Book Selection (i.e. Genesis, Mark, John, etc.)
3. Chapter Number (i.e. chapter number, for John 3:16, the chapter would be 3)
4. Verse Number (i.e. verse number, for John 3:16, the verse would be 16)
5. Verse Text (i.e. text of the actual Bible Verse)
6. Submit Button (will persist all fields in a database table)

Create a Bible Verse Search page with a search form that contains the following form elements:

1. Testament Selection (i.e. Old or New Testament)
2. Book Selection (i.e. Genesis, Mark, John, etc.)
3. Chapter Number (i.e. chapter number, for John 3:16, the chapter would be 3)
4. Verse Number (i.e. verse number, for John 3:16, the verse would be 16)
5. Search Button (will search in the database table for the Bible Verse)

Create a Main Application page with a simple Main Menu (hyperlinks are fine) that allows a user to navigate to the Bible Verse Entry page or the Bible Verse Search page. The application should have a Title and Logo.

The Bible Verse Search Response page should simply display the Bible Verse found in the search or display a “Not Found” error message to the user.

The Main Menu should be available from all pages of the application.

The application should use Layouts with a common Header and Footer.

The application must support a responsive user interface design.

All data entered in all forms must have the proper data types with data validation, and display proper data entry error messages to the user.

Technical Requirements:

1. Must be designed using N-Layer web architecture.
2. Must be designed using a DAO design pattern.
3. Must be designed using OO principles.
4. Must demonstrate IoC and dependency injection.
5. Must demonstrate the use of a logger to log entry and exit for all relevant class methods.
6. All classes must be fully documented including a source file header and header for each class.
7. All class methods must be fully documented including a class header that includes a description of the method, all methods, all return values, and any exceptions thrown.
8. All class method implementation must be fully documented with single inline comments.

**Activity**

This is the last project in this course and is a benchmark assignment that is an individually assessed component. Design and develop the business application using .Net MVC and OO principles that meets the above business requirements.

**Deliverables and Submission (each individual team member):**

A design report document with the following elements:

1. Explanation of the technical approach taken to design and implement the application, including a detailed list of all classes, methods, variables, and beans, as appropriate and relevant to your application. Use UML diagrams as necessary.
2. ER diagram of the database designed.
3. Screens shots of all data entry forms, data entry forms with data validation errors, and a search result screen showing the Bible Version for John 3:16.
4. A technical explanation with 150-250 words on how this application could be extended to include a “Bible Verse of the Day” feature that would be accessible within the application and externally from the application. Provide a block diagram to substantiate your answers and what .NET technologies would be required to support this feature (e.g., diagrams, tables).
5. In addition, write a 150- to 250-word explanation on how knowledge about enterprise-programming tools can solve business problems to provide common good derived from the Christian worldview, based on the experience of creating a Bible verse application.
6. A link to your GIT Repository.
7. Upload the design document to LoudCloud.

Upload all source code used to implement, deploy, and test the application to your GIT account and as a zip file to LoudCloud.

GCU style is not required, but solid academic writing is expected.

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.

***Benchmark Information***

*This benchmark assignment assesses the following programmatic competencies and professional standards:*

*BS Computer Programming*

*1.2: Solve business problems using enterprise-programming tools.*