**.NET Application Programming**

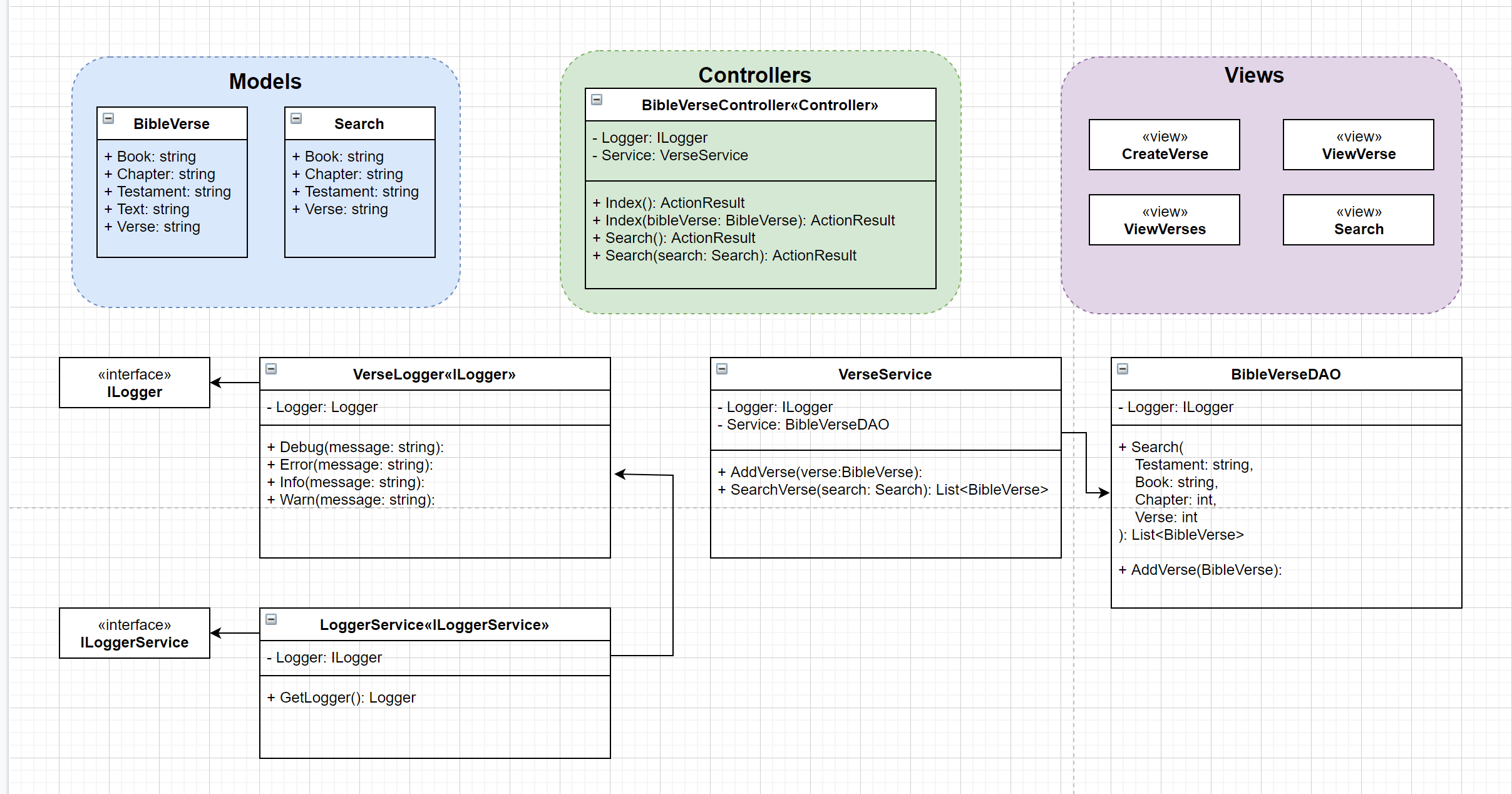
**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | Bible Verse Application | |
| **Date:** | 2/20/2021 | |
| **Revision:** | 1.0 | |
| **Team:** | 1. Richard Williamson | |
|  | |
|  | |
|  | |
| **GIT URL:** | <https://github.com/darthxvaderxd/CST-247-assignments/tree/main/BibleVerseApplication> | |
| **Peer Review:** | *Y* | We acknowledge that our team has reviewed this Report and we agree to the approach we are all taking. |

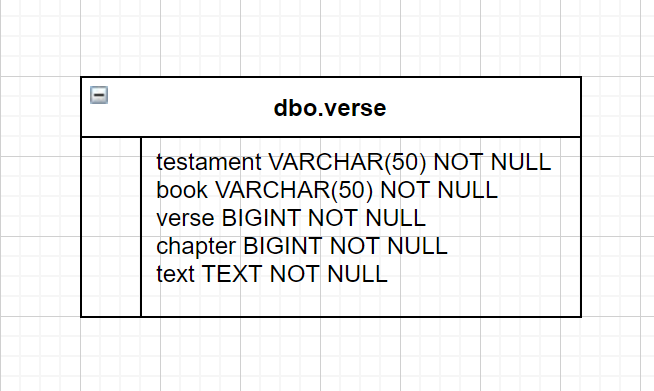
**Design Documentation**

**Install Instructions:** A data base named “BibleVerseRW” will need to be created. The DDL will need to be run in the DDL Folder of the BibleVerseApplication folder which contains all the code.

**UML Diagrams:**



**ER Diagram:**



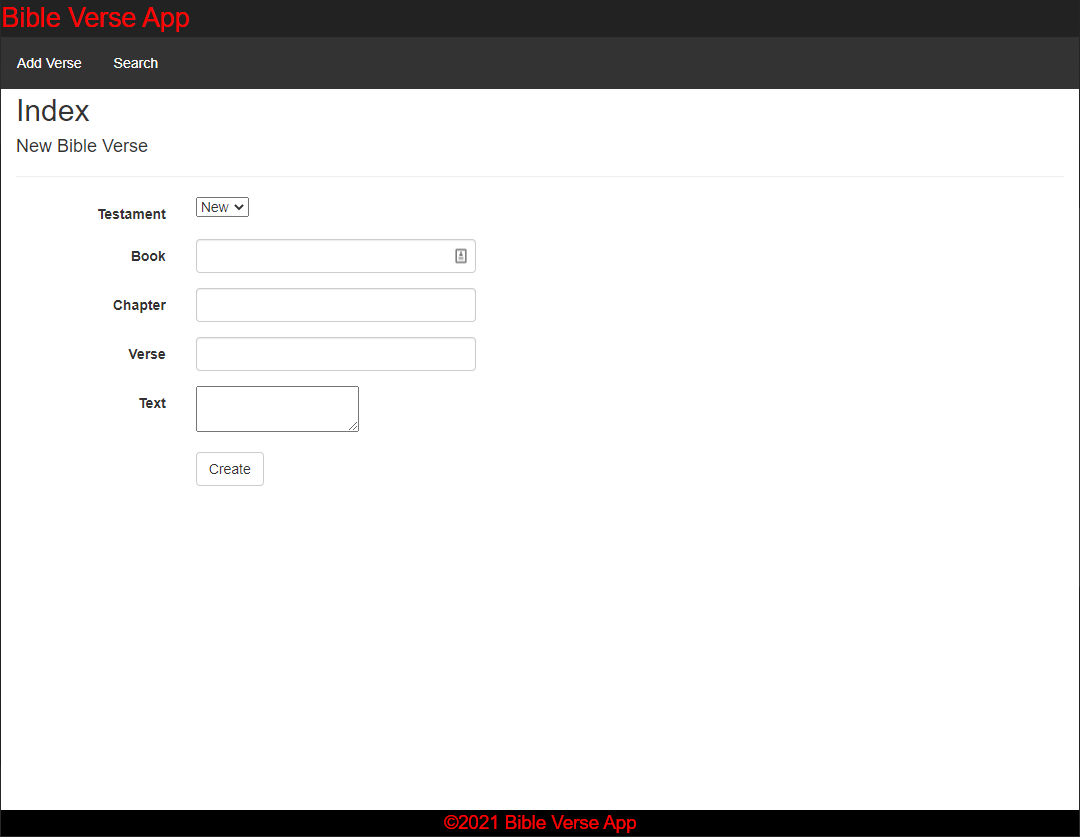
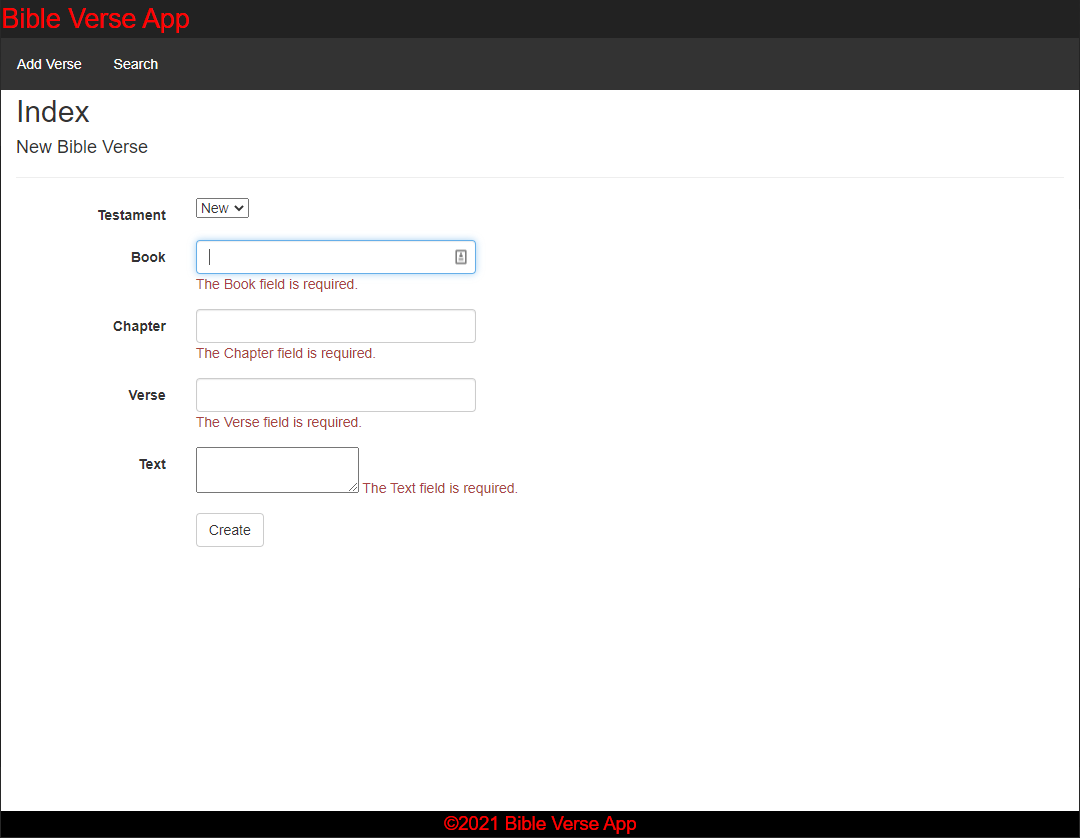
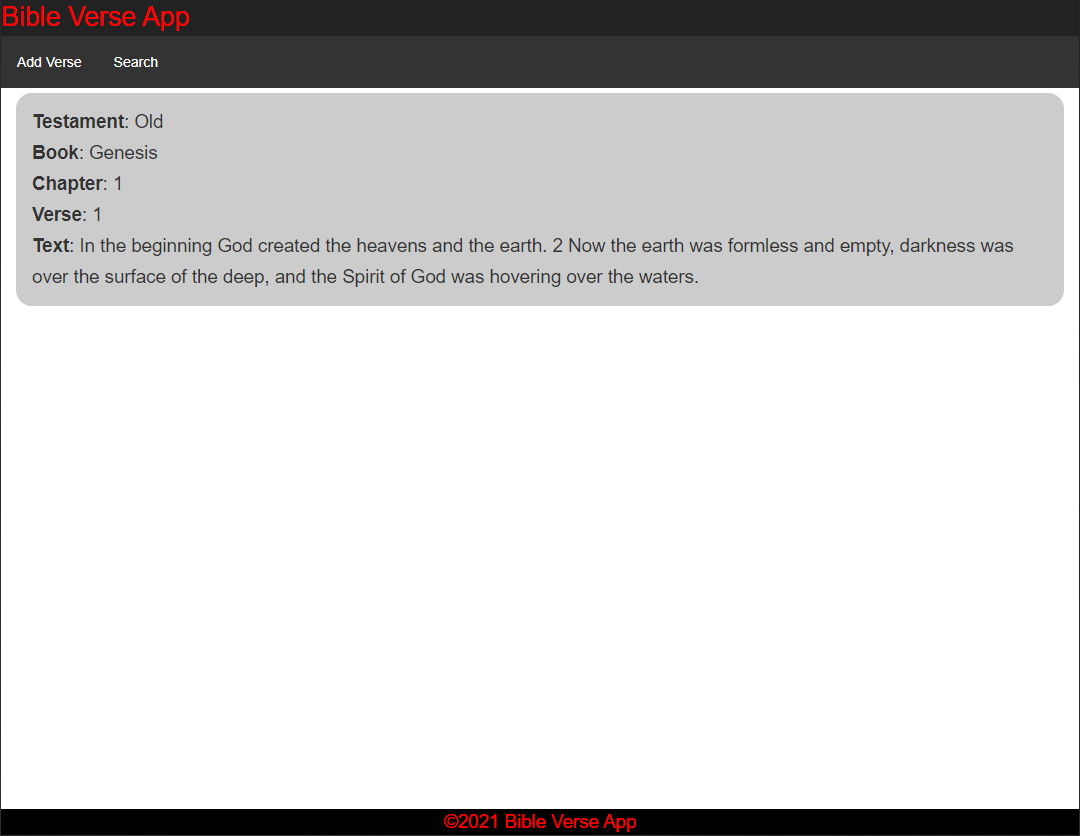
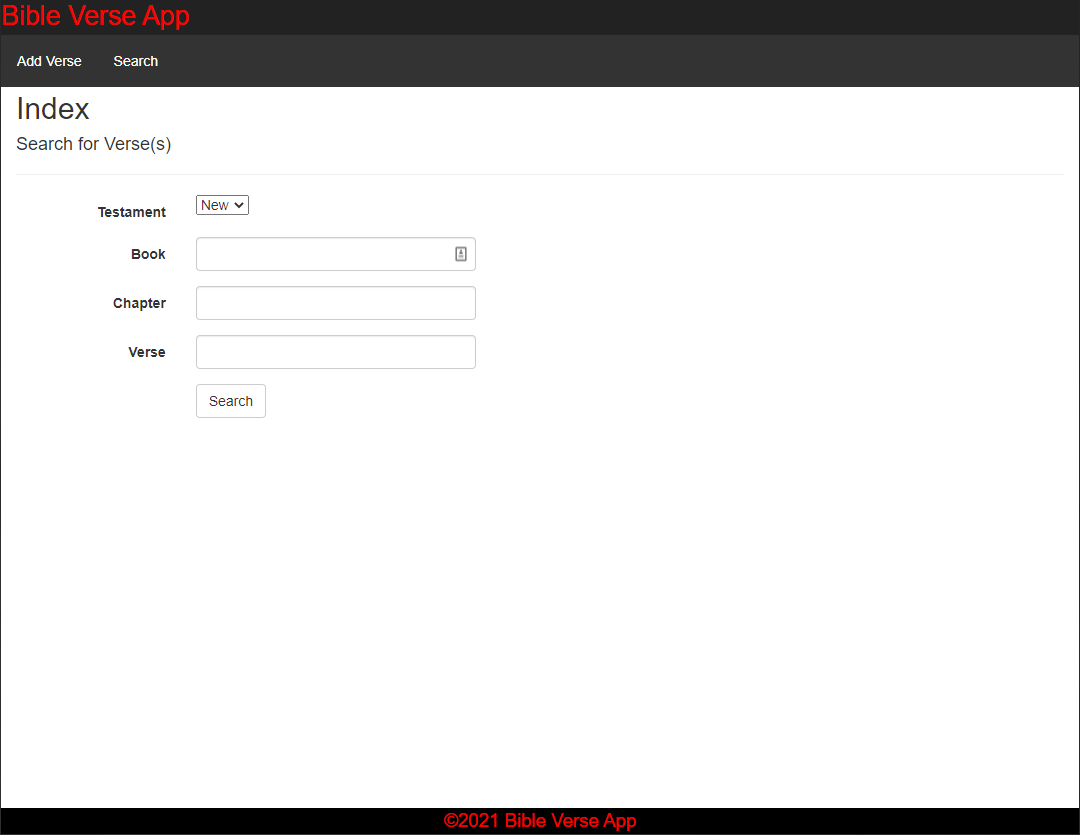
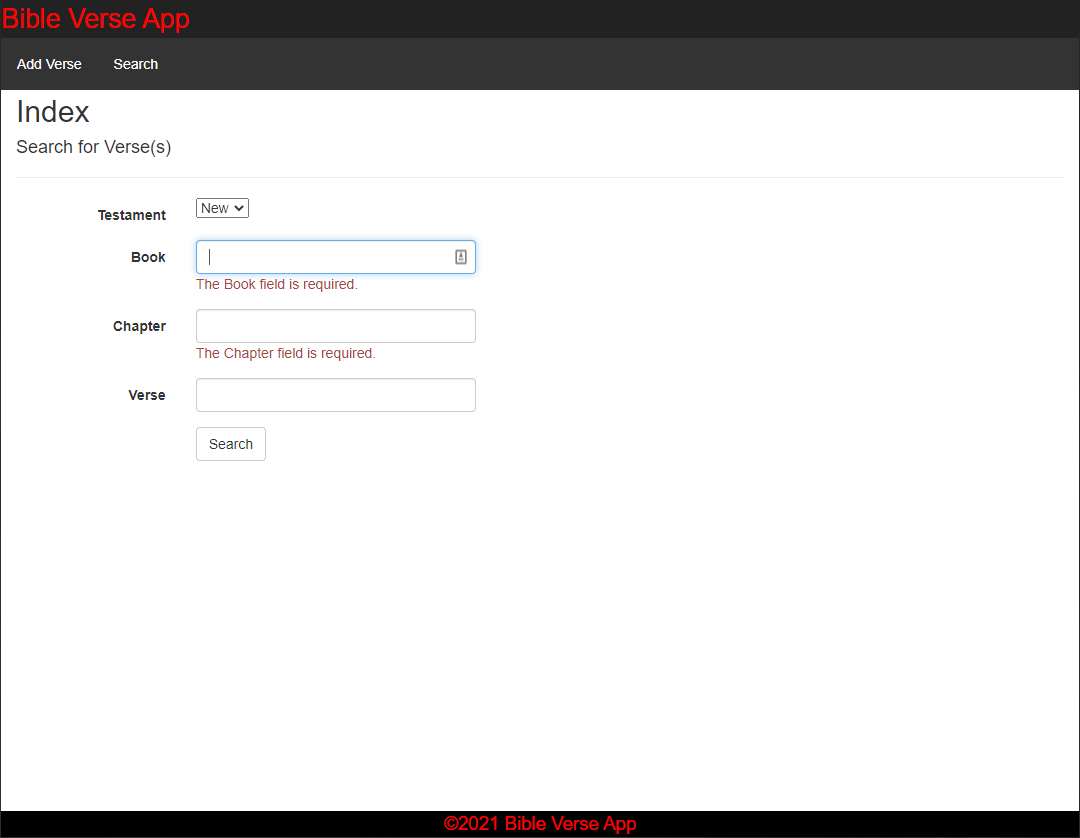
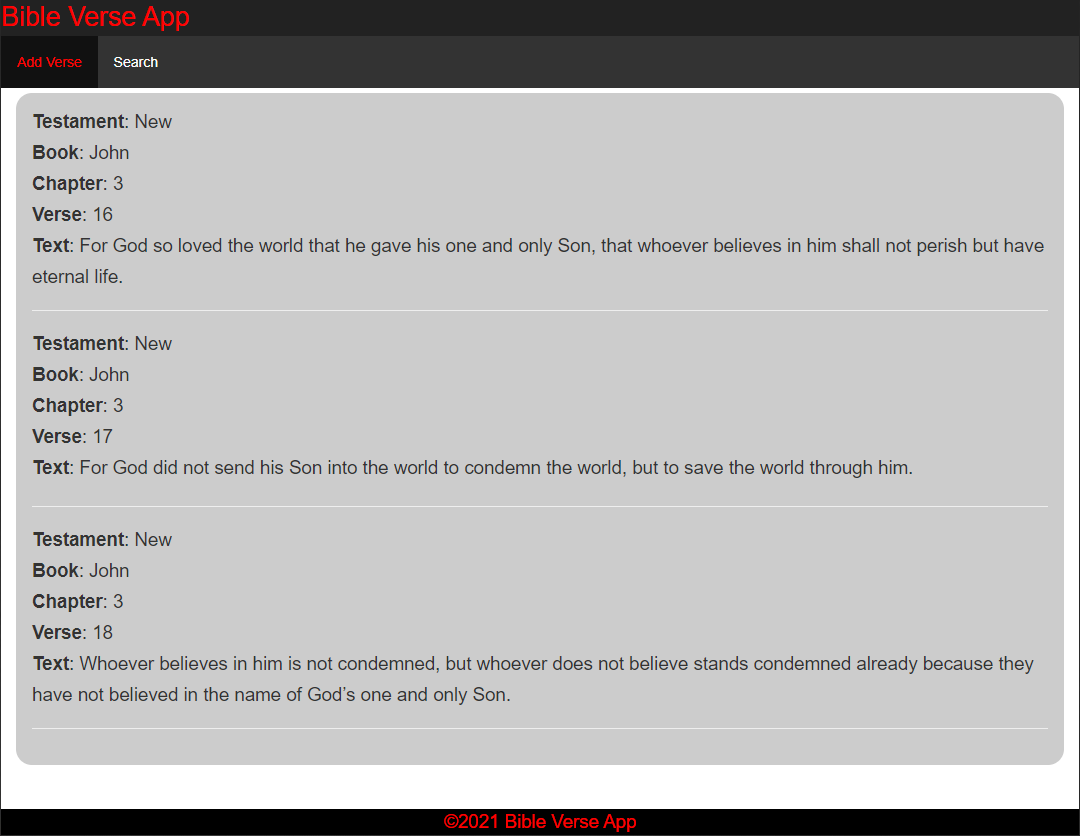
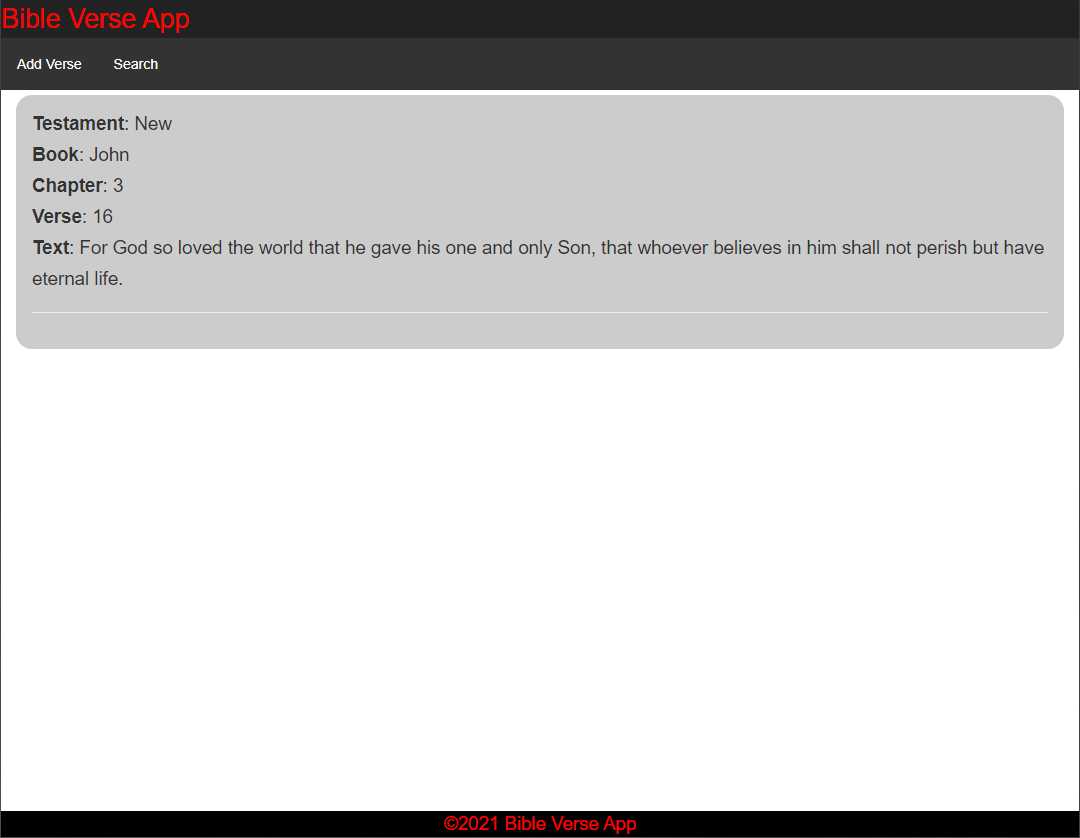
**Technical Approach:**

An ASP.NET Web Application (.NET Framework) project was created to leverage the MVC framework of C#. This allows us to create Controllers, Models and Views for application. This separates the area of concerns and allows us to quickly create an application as requested. There is one controller which is the BibleVerseController and default traffic is routed to the Index action. The Index action returns the CreateVerse view which allows us to create a verse. This is posted to the controller and then saved in the database. The user is then routed to the ViewVerse View. Our user can also search for verses via the Search action on our controller. This posts a Search model containing the users desired search parameters. The user is then routed to the ViewVerses view which shows the search results.

IoC was used for dependency injection of the logging. There is a ILoggingService interface which is implemented by the LoggerService. This class is used to retrieve the Logger to be used throughout the application. There is an ILogger interface which is implemented by the VerseLogger. This is then retrieved from the Util class through the GetLogger function. This allows us to use a different logger based on different situations.

For database, the local SQL Server was leveraged. The database created was BibleVerseRW which has a table dbo.verse. The connection information is retrieved from the Util class via the constant string DB\_CONN. In order to get or store verses in the database there was a VerseService class created to handle sending the request to the database via the BibleVerseDAO class.

**Screenshots:**

1. CreateVerse View upon initial loading.  
   
2. CreateVerse View showing data validation.  
   
3. ViewVerse View after successful creation of new Verse.  
   
4. Search View upon initial loading to perform a search.  
   
5. Search View showing data validation.  
   
6. ViewVerses view after successful search without verse.  
   
7. ViewVerses view after successful search with verse specified.  
   

**How to add Bible Verse of the Day?**

A bible verse of the day feature could be added through addition of a timer that runs every twenty-four hours. When this timer is triggered it randomly chooses a verse for that day. A new boolean column is then added to the verse table called vod. The old entry is updated to be false and the new verse chosen would be updated to be true. Then a new controller action would be added called VerseOfDay. This would fetch the ViewVerse view and pass in the verse of the day. This would be the most basic way for a verse of the day to be implemented. You could take this one step further and create a dbo.vod table which would have book, chapter, verse, and date columns. This would allow you to keep a chronological order of verse of the day which could be archived and later gone through. This would also allow you to ensure the same verse is not chosen within a given time window depending on the number of verses stored in the database.

**How can enterprise programming tools solve business problems to provide common good derived from the Christian worldview, based on the experience of creating a Bible verse application?**

With the enterprise programing tools we can create applications which would benefit mankind such as the bible verse application. This provides a common good in the eyes of a Christian worldview because it allows you to view and store verses from the bible. There are applications that can also help the world such as tracking the needs of villages in third world countries and allowing people to purchase those things for the village. Instead of creating applications for the purpose of making a company money we could focus on creating applications that help mankind over making money for someone.