

Inclusion

CST-391: JavaScript Web Application Development

Andrew Rodriguez

Bobby Estey

September 7, 2025

Inclusion

Inclusion — Bringing the Word Together Across Translations

Inclusion is a Christian-themed personal study web app that allows users to create, read, update, and delete personal study notes (Verse Comments) tied to specific Bible verses and translations. This app helps users engage with scripture, track insights across various translations, and organize their personal notes while maintaining the integrity of the original Bible text.

The main goal of Inclusion is to offer a clear, easy-to-use, and accessible method for personal Bible study, helping users reflect on verses, compare translations, and keep a personal record of their insights.

Functionality Requirements

1. **As a user**, I want to view all Bible verses so I can select a verse to study.
2. **As a user**, I want to search for verses by book, chapter, or keyword so I can quickly find specific scripture.
3. **As a user**, I want to view all comments for a selected verse so I can see others' insights or my previous notes.
4. **As a user**, I want to create a personal comment for a verse so I can record my study notes.
5. **As a user**, I want to edit my comments to update or clarify my insights.
6. **As a user**, I want to delete my own comments so I can remove outdated or incorrect notes.
7. **As a user**, I want to filter comments by translation so I can compare insights across versions like KJV and ACV.
8. **As a user**, I want to view the details of a verse, including book, chapter, and translation, so I have context for my comments.
9. **As a user**, I want to see a list of my own comments in one place so I can review my personal study notes efficiently.
10. **As a user**, I want to sort comments by date so I can see my most recent notes first.
11. **As a user**, I want to navigate easily between verses and comments so I can study without confusion.

Initial Database Design: ER Diagram

Tables and Fields

1. books

id INT, PK, AUTO_INCREMENT
translation VARCHAR(50), NOT NULL
name VARCHAR(50), NOT NULL
testament ENUM('OT', 'NT'), NOT NULL

2. translations

translation VARCHAR(255), PK, NOT NULL
title VARCHAR(255), NULL
license TEXT, NULL

3. verses

id INT, PK, AUTO_INCREMENT
translation VARCHAR(50), NOT NULL
book_id INT, FK → books.id
chapter INT, NOT NULL
verse INT, NOT NULL
text TEXT, NOT NULL

4. VerseComments

id INT, PK, AUTO_INCREMENT
verse_id INT, FK → verses.id
comment_text TEXT, NOT NULL
created_at DATETIME, DEFAULT CURRENT_TIMESTAMP
translation VARCHAR(255), NOT NULL (default: KJV)

Relationships

books.id → verses.book_id (one-to-many)
translations.translation → verses.translation (one-to-many)
verses.id → VerseComments.verse_id (one-to-many)

Initial UI Sitemap

1. Home Page

1.1 Browse Verses

1.1.1 Verse Detail

1.1.1.1 Comments (Create, Read, Update, Delete)

1.2 My Comments

1.2.1 List of personal comments

1.2.2 Edit/Delete options

1.3 About Page

1.4 Admin Page (optional)

1.4.1 Manage Books

1.4.2 Manage Translations

Initial UI Wireframes (Text)

Home Page

- Header: Application Title (“Inclusion”)
- Navigation: Browse Verses, My Comments, About, and Admin
- Main Section: Welcome text and quick link buttons

Browse Verses Page

- Dropdowns: Select Translation, then Select Book, then Select Chapter, and finally Select Verse.
- Button: “View Verse”
- Below: Display the Verse Text.
- Comments Section: List of Comments, plus an “Add Comment” button.

Verse Detail Page

- Verse text in large font
- Comment box (textarea + submit)
- Existing comments with Edit/Delete buttons

My Comments Page

- List of users’ comments (table or cards)
- Actions: Edit, Delete

About Page

- Static text: Purpose of the app, Christian theme inspiration

Initial UML Classes (Text Version)

Book

- Attributes: id: int, translation: string, name: string, testament: enum(OT, NT)
- Methods: getBooks(), getBookById(id)
- Relationships: 1 Book → many Verses

Translation

- Attributes: translation: string, title: string, license: text
- Methods: getTranslations(), getTranslationById(id)

Verse

- Attributes: id: int, translation: string, book_id: int, chapter: int, verse: int, text: string
- Methods: getVersesByBook(book_id), getVerseById(id)
- Relationships: 1 Verse → many VerseComments

VerseComment

- Attributes: id: int, verse_id: int, comment_text: text, created_at: datetime, translation: string
- Methods: addComment(c), updateComment(id, c), deleteComment(id), getCommentsByVerse(verse_id)

CommentService

- Methods: createComment(data), readComment(id), updateComment(id, data), deleteComment(id), listComments(verse_id)

ApiController

- Routes: /books, /translations, /verses, /comments

Risks

1. **Database Schema Changes:** Future modifications to the Bible verse or translation tables could potentially disrupt CRUD operations or necessitate updates to the API.
2. **Data Integrity:** Ensure that user comments are accurately linked to verses and translations, avoiding duplication or loss.
3. **Front-end Integration:** The Angular and React versions of the app may exhibit varying behaviors when consuming the same API, potentially leading to inconsistencies.
4. **User Authentication:** If implemented later, authentication could introduce additional complexity for comment creation, editing, and deletion.
5. **Scalability:** A large number of verses (approximately 500,000+) may slow down searches or comment retrieval.
6. **Cross-Translation Compatibility:** Comparing and displaying comments across multiple translations may necessitate additional logic and meticulous UI design.
7. **Deployment & Environment Differences:** Variations between development, testing, and production environments could cause runtime errors (Node version, TypeScript configuration, MySQL setup).
8. **Time Constraints:** Completing both Angular and React front-ends and back-end services may pose challenges within project deadlines.