Simplifying reference management, bringing JabRef and LibreOffice closer than ever.

Improved JabRef-LibreOffice Integration

GSoC 2025 Proposal

Priyanshu Gupta

Table of Contents

About me

Greetings! I'm Priyanshu Gupta, currently pursuing Bachelors in Computer Science Engineering at Sharda University, India.

My open source journey began in December 2024 when I started contributing to JabRef. Through this experience, I've gained invaluable insights into the practical application of Java in real-world projects. I gained knowledge about testing, implementing new features, Gradle usage, test case development, and more.

Contributing to JabRef has not only made me better at coding, but I also got to meet some amazing folks of the community and got to learn so much from them.

Some details and links about me:

Email	priyanshu16095@gmail.com
Github	priyanshu16095
LinkedIn	priyanshu16095
Portfolio Website	priyanshu-gupta.vercel.app
University	Sharda University, Greater Noida
Course	B.Tech in Computer Science
Timezone	Indian Standard Time (IST), UTC +5:30

Background

As a second-year student in engineering I am comfortable with the fundamentals of computer science, operating systems, computer networks and more.

I know Java, JavaScript, and software design principles like design patterns and SOLID principles.
I have developed many frontend, backend, and full-stack projects for my portfolio, including a university project, Secura, a React.js based frontend for entry permission software. I also have basic knowledge of DevOps.

My introduction with JabRef and familiarity with BibTeX and BibLateX.
My first introduction to JabRef was through code, but now I understand it from both the user and developer perspectives.
I am familiar with BibTeX and BibLaTeX and gained deeper knowledge of them while contributing to JabRef.
BibTeX is a reference management tool used for formatting bibliographies in LaTeX documents and storing reference data in a .bib file, while BibLaTeX is a more modern alternative to BibTeX that is more customizable and provides more control over formatting.

Have I participated in GSoC before?
No, this will be my first time participating in GSoC.

Availability

I would be able to devote approximately 50 hours every week to GSoC. During May I will have my end-sessional examinations, and time devoted would be around 2-3 hours per day for that time.

Apart from examinations, I have no other obligations and will devote 100% of my time to GSoC.

Synopsis

JabRef is an open-source, cross-platform citation and reference management software. It uses BibTeX and BibLaTeX as its native formats, making it popular among LaTeX users.

Being a feature packed software, one of the most appreciated features is JabRef's integration with LibreOffice, which is actively developed and has got the support for CSL styles in GSoC 2024.

Seeing that growth and support this feature has received, JabRef has decided to expand further by adding support for BST styles via UI, adding support for custom CSL styles, improvement in support for CSL styles and the cross-compatibility with other reference management software.

All these features come with their own strengths:

JabRef can stand out easily by adding support for BST styles via UI in LibreOffice as other software rely on CSL styles, whereas JabRef's native format is BibTeX/BibLaTeX. It will greatly benefit for LaTeX users who also work with LibreOffice documents.
Cross-compatibility will eliminate software-lock in and allows users to seamlessly collaborate with those using other software.
Improving support for CSL styles will make JabRef a strong choice for users who rely on CSL styles and effectively address their needs. Users also look forward to improvements in CSL styles and consider it a useful feature in their work [Reference].
Adding support for custom CSL styles JabRef will match the flexibility provided by other software, ensuring that diverse requirements are met.

These features will empower students and researchers, improve versatility and also make JabRef a top choice for users.

Project description

The JabRef-LibreOffice integration is a feature that enhances the reference management capabilities of JabRef by allowing users to cite library entries directly into LibreOffice documents and generate bibliographies based on those citations.

The current implementation supports custom styles (JStyles) and CSL styles, but there are several areas where the integration can be improved to provide a more seamless and user-friendly experience.

Below are the key projects that aim to enhance the JabRef-LibreOffice integration:

BST Style Support	Link
Currently, JabRef supports BST styles, but this functionality is not accessible through the user interface (UI). The goal of this project is to enable users to select a .bst file via the UI, which will then be used for rendering citations and bibliographies in LibreOffice documents. This will make it easier for users who rely on BST styles for their LaTeX-based workflows to integrate with LibreOffice.	
Project Deliverables: ● Implement UI support for selecting .bst files. ● Ensure that the selected .bst file is used for rendering citations and bibliographies in LibreOffice documents. ● Address any related issues or bugs in the BST style implementation.	

Improved Support for CSL Styles

A) Footnote-based citation support for CSL styles Link

Currently, inserting CSL-style citations in footnotes can lead to unexpected behavior, especially with numeric citation styles. A proper definition of the global order of citations is required to ensure correct numbering and bibliography formatting.

Project Deliverables:

- Implement a consistent global citation order for CSL styles, ensuring correct numbering and bibliography formatting in footnotes.
- Adapt the existing JStyle solution for footnotes to work seamlessly with CSL styles.
- Verify the functionality and the issue reported by a user.

B.) Support for custom CSL Styles Link

Many users prefer to use custom-designed CSL style files that are not part of the official Zotero-CSL repository. This feature will enhance JabRef to allow the import and usage of custom CSL files.

Project Deliverables:

- Enable users to load and apply external CSL style files in JabRef.
- Implement a user-friendly interface for managing and selecting custom CSL styles.
- Test the functionality using custom CSL styles.

Cross-compatibility with other reference management software	Link
JabRef currently uses a custom internal format for managing references in LibreOffice. In contrast, reference management tools like Zotero and Mendeley use a standardized format based on CSL JSON, allowing them to read each other's citations seamlessly. This lack of standardization in JabRef creates a compatibility barrier, making it difficult for users to work interchangeably. To address this issue, JabRef should adopt the CSL JSON format used by Zotero and Mendeley, ensuring interoperability.	
Project Deliverables: ● Implement support for the CSL JSON citation format used by Zotero and Mendeley. ● Verify the compatibility with Zotero and Mendeley.	

Seamless Citation Style Type Switching
Currently, JabRef in LibreOffice does not support automatic updates when switching between CSL-based formats and JStyle (or BST)-based formats. If a user needs to switch between citation style types, such as a different CSL style, they must manually re-cite all entries and refresh the bibliography, which can be time-consuming and inconvenient. The goal of this project is to enable seamless switching between different citation style types without requiring manual re-citation.
Project Deliverables: ● Implement a unified "reference mark" format to allow citation data to be parsed consistently across the families of CSL, BST, and JStyle formats. ● Ensure that references in LibreOffice documents automatically update when the citation style is changed.

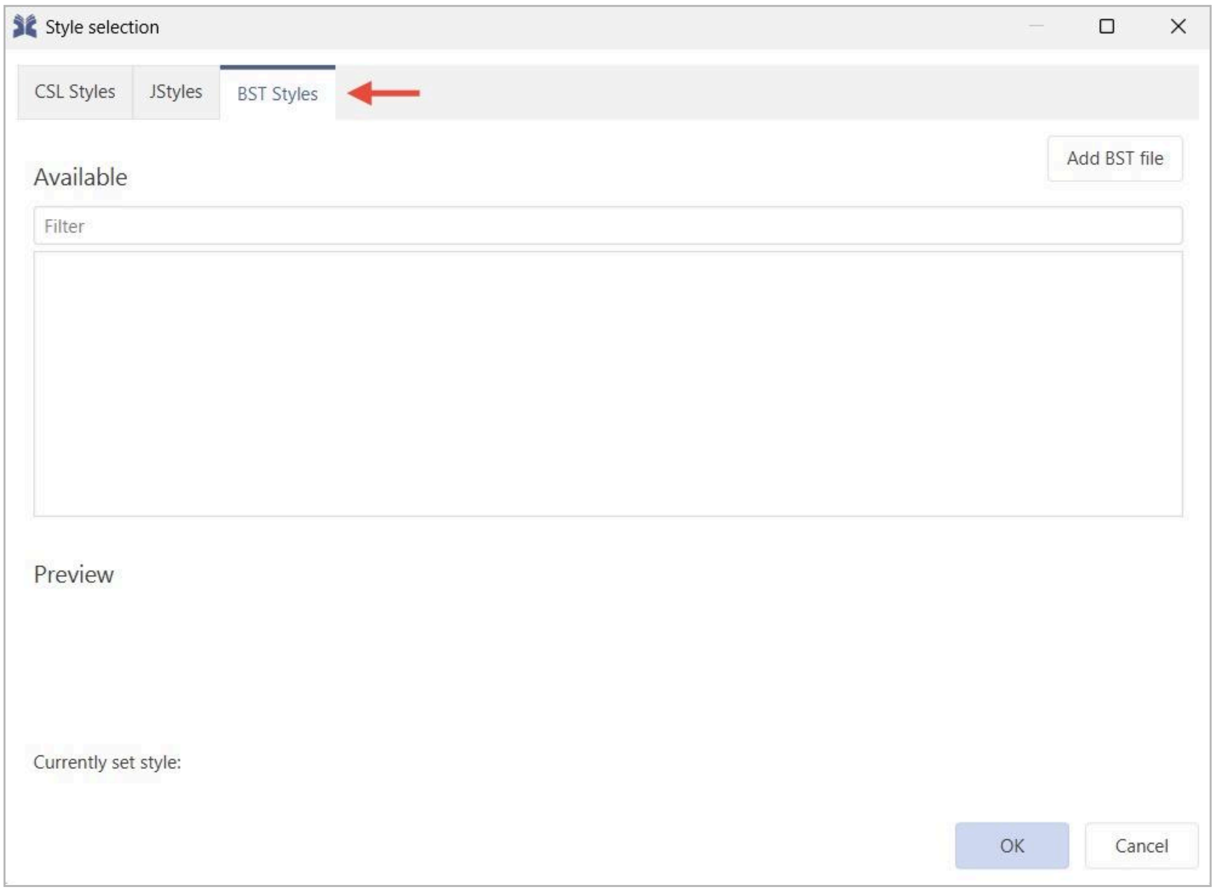
How can BST Style Support be Implemented?

JabRef already supports the BST style for citation, but this functionality is not accessible through the user interface (UI), specifically in the **Style Selection** dialog for **LibreOffice/OpenOffice panel**.

Just like the UI for CSL and JStyles is present, a **third tab** needs to be added to that dialog, making it available for use in LibreOffice/OpenOffice.

The added **BST Styles** tab would display the available styles, a button to load BST files, provide a filter for searching, include a section to preview styles, and show the currently set style at the bottom.

As shown:



The available BST styles will be displayed in a **ListView** in the **FXML** file as:

```
<Tab text="BST Styles">
    <VBox spacing="10.0" VBox.vgrow="ALWAYS">
        <padding>
            <Insets top="10.0" right="10.0" bottom="10.0" left="10.0"/>
        </padding>
        <VBox spacing="4.0" VBox.vgrow="ALWAYS">
            <HBox alignment="BASELINE_CENTER">
                <Label text="%Available" styleClass="sectionHeader"/>
                <Region HBox.hgrow="ALWAYS" />
                <Button text="Add BST file" alignment="CENTER_RIGHT" />
            </HBox>
            <CustomTextField fx:id="searchBox" promptText="%Filter" prefHeight="20.0"/>
            <ListView fx:id="availableListView" VBox.vgrow="ALWAYS"/>
        </VBox>
        <VBox spacing="4.0">
            <Label text="%Preview" styleClass="sectionHeader"/>
            <VBox fx:id="bstPreviewBox" prefHeight="300.0" spacing="4.0"/>
        </VBox>
    </VBox>
</Tab>
```

The functionality to select a BST style file can be added as:

```
@FXML
    private void selectBstFile(ActionEvent event) {
        FileDialogConfiguration fileDialogConfiguration = new
                FileDialogConfiguration.Builder()
                .addExtensionFilter(StandardFileType.BST)
                .withDefaultExtension(StandardFileType.BST)
                .withInitialDirectory(preferences.getFilePreferences().getWorkingDirectory())
                .build();

        dialogService.showFileOpenDialog(fileDialogConfiguration).ifPresent(bstFile ->
              viewModel.addBstStyle(bstFile));
    }
```

To preview the added BST styles in the tab, the functionality is already available in the file **BSTPreviewLayout.java**, located in the **org.jabref.logic.bst** directory.

After adding and selecting the file, the currentStyle variable, which is an instance of a class implementing the **OOStyle interface** in the file **OpenOfficePreferences.java**, is updated.

```java
if (currentStyle instanceof JStyle jStyle) {
    try {
        jStyle.ensureUpToDate();
    } catch (IOException ex) {
        LOGGER.warn("Unable to reload style file '{}'", jStyle.getPath(), ex);
        String msg = Localization.lang("Unable to reload style file")
                     + " '" + jStyle.getPath() + "'\n"
                     + ex.getMessage();
        new OOError(title, msg, ex).showErrorDialog(dialogService);
        return FAIL;
    }
}
```

The buttons for different operations in the OpenOffice/LibreOffice panel function correctly because the **getOrUpdateTheStyle()** method is called in the **pushEntries()** method, ensuring that the value of the selected style is up to date.

The **BstVM.java** class is used by **BstPreviewLayout.java** as a BST interpreter that processes **.bst files** to format bibliographic entries into structured reference lists.It uses ANTLR to process BST files, applies style rules through a visitor, and stores runtime data in **BstVMContext record**.

A possible enhancement could be refactoring all the modeling of a BST style in Java into a separate class, **BSTStyle.java**, similar to JStyle and CSL style, to manage variables, file state, and other related functionalities.

Since BST styles are complex, the project aims to provide basic support for them. Once the functionality for rendering BST-style citations in XTextDocument is completed, we can proceed with implementing this issue based on the available time.

How can Support for CSL Styles be Improved?

A.) Footnote-based Citation Support for CSL Styles

Approach JabRef uses for ordering of citation groups:	Reference
The ordering of citations in a document is determined at two levels: ● Local Order: Order within each citation each group. ● Global Order: Order of citation groups in text.	
Sorting approach: ● Replaces the textranges of footnote citations with the textranges of footnote marks. ● Retrieves marks position. ● Sorts citation top-to-bottom, left-to-right.	

JStyles already includes logic for numeric citation markers in **JStyleGetNumCitationMarker.java**, but it needs to be refactored to support **footnote-based CSL formatting**.

A **global order** mechanism must be defined for footnote citations to ensure consistent numbering across citations. The citation processing methods will be modified to track and update footnote citations correctly.

To verify correctness, unit and integration tests will be implemented for different numeric styles. The implementation will be validated with real-world footnote citation scenarios in LibreOffice.

B.) Support for Custom CSL Styles

JabRef needs an addition in the UI to allow users to import external CSL style files.

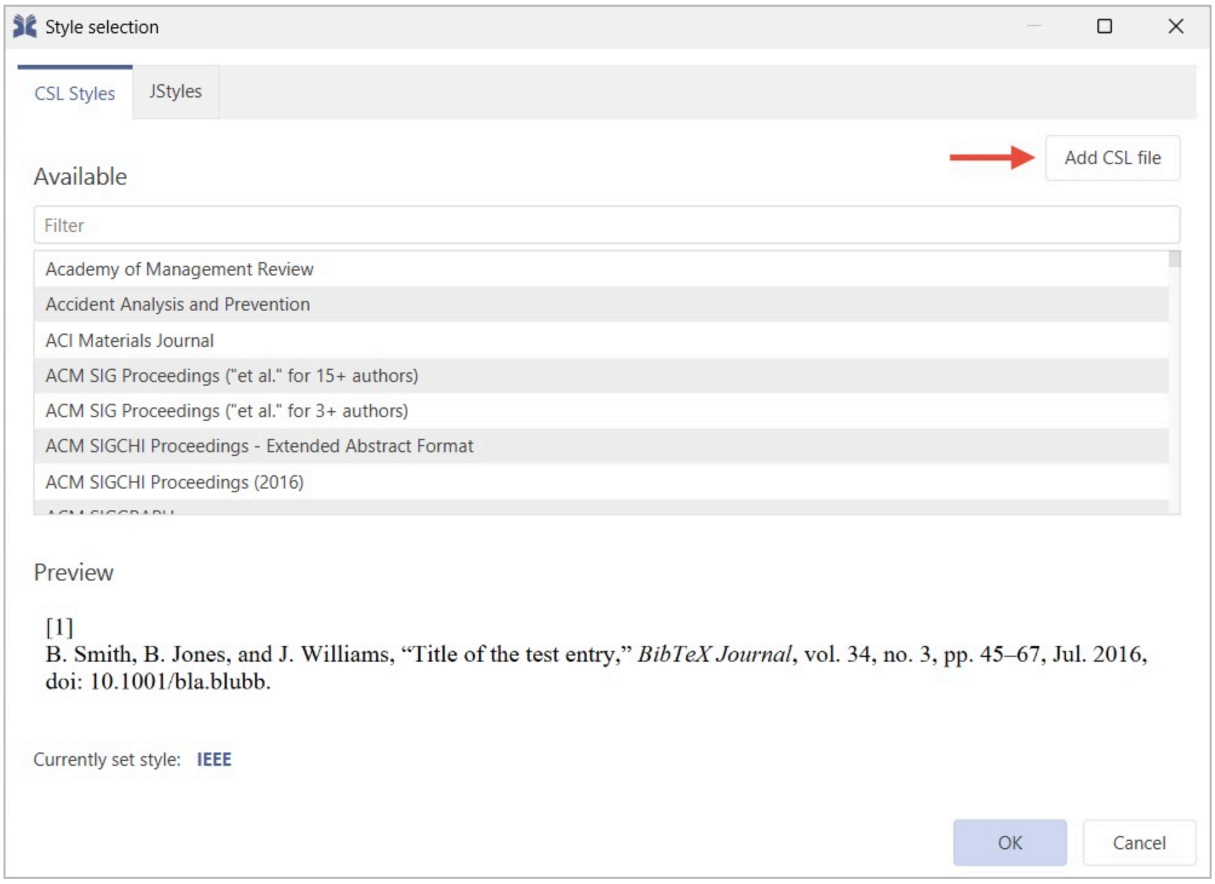
In the **CSL Styles tab**, an **Add CSL Styles** button is added, which, when clicked, displays a **File Picker Dialog**.

The CSL processor will dynamically load and validate custom styles.

The added custom CSL styles update the **selectedLayoutProperty** variable in the view model class with a single click, which in turn updates the preview text. On a double click, they call the **handleCslStyleSelection()** method in the view model class, updating the currentStyle variable in **OpenOfficePreferences**.

Test cases will be written to verify the correct loading and application of custom CSL styles.

The **Add CSL file** button for loading custom CSL files will be added to the **CSL Styles tab**, as shown:



Academy of Management Review
Accident Analysis and Prevention
ACI Materials Journal
ACM SIG Proceedings ("et al." for 15+ authors)
ACM SIG Proceedings ("et al." for 3+ authors)
ACM SIGCHI Proceedings - Extended Abstract Format
ACM SIGCHI Proceedings (2016)

[1]
B. Smith, B. Jones, and J. Williams, "Title of the test entry," *BibTeX Journal*, vol. 34, no. 3, pp. 45–67, Jul. 2016, doi: 10.1001/bla.blubb.

Currently set style: IEEE

How can Citation Style Type Switching be Implemented?

LibreOffice currently supports reference marks for citations through the **CSLReferenceMark** and **ReferenceMark** classes and automatically updates when the citation style is changed, such as from JStyle to CSL styles. However, there is no unified way to switch between different families of styles.

Identify Existing Reference Marks
LibreOffice's UNO API will be utilized to retrieve all reference marks within the document.

The **NamedRangeReferenceMark** class implements the **NamedRange** interface to represent and manage reference marks in LibreOffice documents

Parse Reference Marks into a Unified Format
To ensure uniformity, reference data will be stored in **JSON/XML format** within LibreOffice document annotations. Citation keys, page numbers, and metadata will be extracted and converted into a structured format.

This is related to the cross-compatibility project and it should already be implemented.

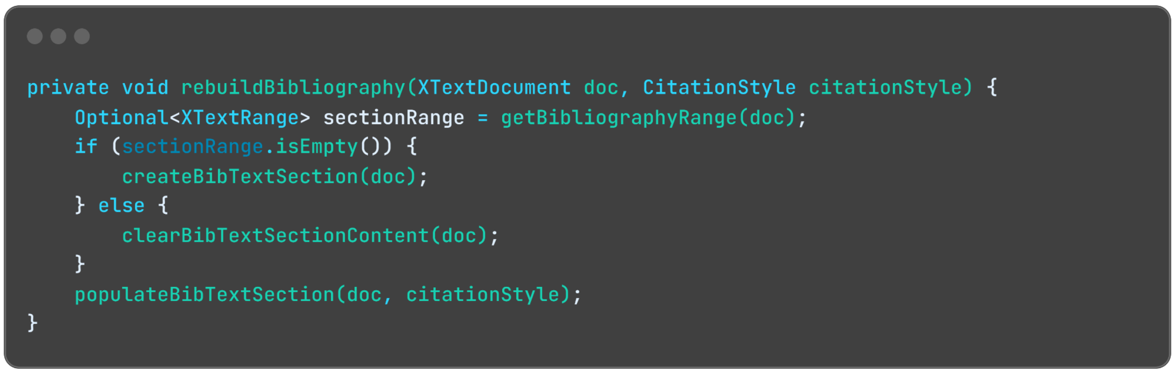
Apply the New Citation Style
The existing functionality, the **Style select dialog** allows users to apply the formats from JabRef.

Update Document with New Citations
Keeping the reference marks in place, citations will be replaced by newly formatted citations, generated by their respective citation text generators, based on the style.

Refresh Bibliography

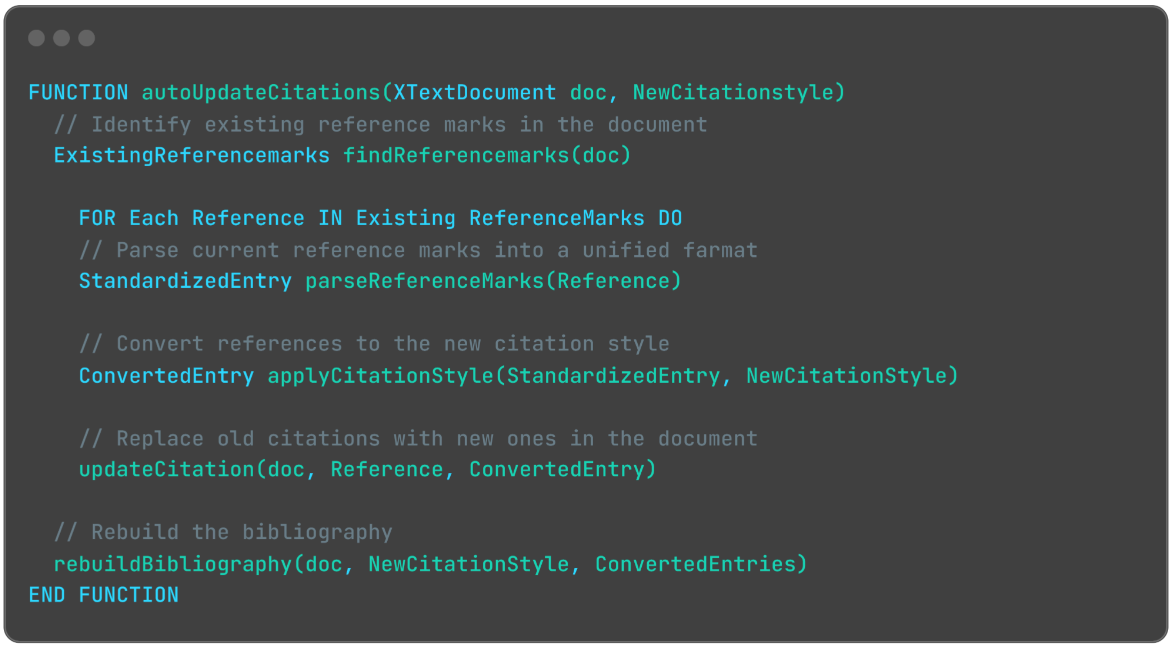
To maintain consistency between in-text citations and the bibliography, a new citation processing mechanism will be implemented. This ensures real-time updates when changes are made.

The bibliography update is handled by:



```
private void rebuildBibliography(XTextDocument doc, CitationStyle citationStyle) {
    Optional<XTextRange> sectionRange = getBibliographyRange(doc);
    if (sectionRange.isEmpty()) {
        createBibTextSection(doc);
    } else {
        clearBibTextSectionContent(doc);
    }
    populateBibTextSection(doc, citationStyle);
}
```

The pseudocode can provide an overview of this, as shown:



```
FUNCTION autoUpdateCitations(XTextDocument doc, NewCitationstyle)
  // Identify existing reference marks in the document
  ExistingReferencemarks findReferencemarks(doc)

    FOR Each Reference IN Existing ReferenceMarks DO
    // Parse current reference marks into a unified farmat
    StandardizedEntry parseReferenceMarks(Reference)

    // Convert references to the new citation style
    ConvertedEntry applyCitationStyle(StandardizedEntry, NewCitationStyle)

    // Replace old citations with new ones in the document
    updateCitation(doc, Reference, ConvertedEntry)

  // Rebuild the bibliography
  rebuildBibliography(doc, NewCitationStyle, ConvertedEntries)
END FUNCTION
```

In the end, it would be tested by switching between types of CSL, JStyle, and BST styles in various document structures.

How can Cross-Compatibility be Implemented with other Reference Management Software?

Understanding Zotero's Implementation
Zotero uses RTF/ODF-Scan which is an add-on that allows users to insert plain-text citation markers in documents created with any word processor. These markers can later be converted into active Zotero citations after saving the document in the OpenDocument Text (.odt) format. This enables Zotero citation support in word processors without dedicated plugins (Zotero ODF-Scan).

Citation Marker Format used by Zotero consist of five sections:
- Citation prefix
- Author and year
- Locator (e.g. page number)
- Citation suffix
- Unique item identifier (e.g. item ID)

Example:
{Prefix | Author, (Year) | Locator | Suffix | Unique Zotero ID}

In contrast, JabRef stores only name, citation keys, citation numbers and unique ID.
For further reference, see this comment.

Adopting the CSL JSON Format for Citations
Zotero and Mendeley can interchange citations in LibreOffice because they use the **Citation Style Language (CSL)** and **OpenDocument format (ODF) citation markers**. To achieve similar compatibility, JabRef needs to modify its internal citation format to store references in CSL JSON instead of the existing custom format.

JabRef's citation handling will be updated to ensure that citations in LibreOffice follow the structure expected by Zotero and Mendeley. This will allow seamless collaboration between different reference management tools within the same document.

Developing a Converter for Existing JabRef References
A converter will be implemented to migrate existing JabRef citations from the old format to CSL JSON. This converter will be added into OpenOffice, similar to the existing **JabRef_LibreOffice_Converter**, to facilitate a smooth transition.

The role of JabRef_LibreOffice_Converter is to convert reference marks into code if the LibreOffice Writer document is saved in a format other than OpenDocument (.odt), as other formats cause the loss of reference marks.

Ensuring Compatibility
Documents having citations from Zotero and Mendeley should be used with JabRef, to ensure the correct interpretation.

Tests will be written to confirm that LibreOffice documents contain citations from different software functions without formatting errors.

Timeline

Week	**From**	**Till**	**Task**
Week 1	May 08	June 01	**Community bonding period**
Start coding for the project. Time will be utilized to analyze the existing code and APIs.			
Week 2	June 01	June 09	Implement UI to support BST styles.
Week 3	June 09	June 16	Provide the logic to support the UI for BST styles.
Week 4	June 16	June 23	Write tests to verify the functionality.
Support for BST styles implemented.			**Milestone**
Week 5	June 23	June 30	Create an UI for importing custom CSL styles.
Week 6	June 30	July 07	Provide an implementation for processing imported custom CSL styles.
Week 7	July 07	July 14	Write tests to verify the functionality.
Support for custom CSL styles implemented			**Milestone**
Week 8	July 14	July 21	Deeply analyze the implementation of numeric citation markers for JStyle.
Week 9	July 21	July 28	Add implementations like JStyles for CSL styles
Week 10	July 28	Aug 04	Verify the functionality.

No unexpected behavior with CSL style footnote-based citations.			**Milestone**
Week 11	Aug 04	Aug 11	Implement converter for migrating citations.
Week 12	Aug 11	Aug 18	Ensure compatibility with Zotero and Mendeley.
Week 13	Aug 18	Aug 25	Write tests to verify the functionality.
Cross-compatibility with other reference management software implemented.			**Milestone**
Week 14	Aug 25	Sep 01	Unify the reference mark format for all style families.
Week 15	Sep 01	Sep 08	Modify the functionality for identifying existing reference marks and parsing them.
Week 16	Sep 08	Sep 15	Add the code for the automatic updating of citations.
Week 17	Sep 15	Sep 22	Verify the functionality in LibreOffice.
Entries get automatically re-cited if citation style is changed.			**Milestone**
Week 18	Sep 22	Sep 29	Refactor and optimize the implemented code.
Week 19	Sep 29	Oct 06	Write documentation and blog posts for the features. Prepare the final project report.
Week 20	Oct 06	Oct 13	Final Evaluation

Contributions so far

Link	**Description**
Features Implemented	
#12433	GUI for Bibliography Consistency Check
#12374	Feature to Include/Exclude cross Referenced entries when Copying
#12475	CLI for Bibliography Consistency Check
#12516	Dropdown Menu for Citation Key Patterns
#12580	Option for Generating Compound Patterns for Citation Key
#12395	Tab Bar Visibility Toggle functionality
#12495	Actions to Resolve Integrity Issues
#12624	Tabular UI for Linked Files Name
#12574	Keyboard Shortcuts for Fetching Bibliographic Data using Identifier
#12563	Add Rename Subgroup Option
#12784	Functionality to Modify CSL Bibliography.
Enhancements to Implemented Features	
#12539	Fix the Dropdown UI Issue
#12529	Remove Redundant Columns from Consistency Check Table
#12491	Code Enhancements for GUI for Consistency Check
#12459	Fix Displayed Name of Untitled Database
#12503	Fix Usage of “Copy to” option preferences
#12658	Select the Respective Field in Entry Editor
View all my PRs on GitHub.	

Issues I Have Opened

Link	**Description**
#12350	Ctrl+W Does Not Behave as Expected When Opening JabRef
#12501	Add Preference for “Copy To” Option to Include or Exclude Cross-Reference Entries
#12486	Improve UX for “Copy To” Option With Success Dialog
#12500	“Copy To” Option Does Not Follow User Preferences
#12456	Blank Space Displayed in “Copy To” Context Menu When Database Is Untitled

Pull Requests Reviewed

Link	**Description**
#12699	Remove Redundant “Citation Key” Column
#12696	Leave Out Columns with Uniform Value

Overall Contribution

Total PRs	29
Total Issues Raised	5

Why me?

I believe that I have developed the skill of analyzing implementation and working with the codebase while contributing. I always try not to let anyone run behind the implementation or guide me with the basic things.

I have successfully implemented entirely new features multiple times since I was a beginner.

In my first PR/feature, Copy to, I demonstrated that I could handle the logic well, utilized the existing duplicate handler, and successfully implemented the feature.

Another major feature I worked on was the GUI for the Consistency Check, which gave me experience in the full feature development cycle of JabRef. This project required me to deeply analyze the bibliography consistency check implementation, understand how it outputs data, and then manage that data for use in the UI.
This feature showcased my ability to effectively handle both logic and UI development and to work with the existing codebase.

I feel ready to take on these bigger challenges because I have delved into JabRef's OpenOffice/LibreOffice code, having implemented the feature to add custom bibliography titles and formatting to CSL bibliographies.

Through numerous contributions and experience across the full feature development cycle, I also wrote a blog post about one of my features, the Consistency Check GUI.

Apart from that I have also been involved in creating good first issues for newcomers, guiding them to complete them, and reviewing others pull requests.

Future Work

Contributing to JabRef has given me invaluable experience in large-scale Java development, testing, documentation, and open-source collaboration. I've gained hands-on experience with unit testing and community-driven development.

Even after GSoC, I plan to continue contributing by making high-quality pull requests, assisting newcomers, and helping maintainers. I also have future ideas to enhance JabRef and look forward to discussing them.

I would be grateful to the JabRef community and excited to continue learning and contributing.

Thank You 🙂