GSOC 2025 Proposal

JabRef_ev

Improved JabRef-LibreOffice Integration

Name	Priyanshu Gupta
Email	priyanshu16095@gmail.com
Github	priyanshu16095
LinkedIn	priyanshu16095
Portfolio Website	priyanshu-gupta.vercel.app
Timezone	Indian Standard Time (IST), UTC +5:30
Project Size	Large
Mentors	Siedlerchr, subhramit

Table of Contents

About me

Greetings! I'm Priyanshu Gupta, currently pursuing 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.

Availability

I would be able to devote approx 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.

Project description

The LibreOffice-JabRef 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 LibreOffice-JabRef 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.	

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.	

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 change citation styles, 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 CSL, BST, and JStyle formats. • Ensure that references in LibreOffice documents automatically update when the citation style is changed.

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.	
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.	

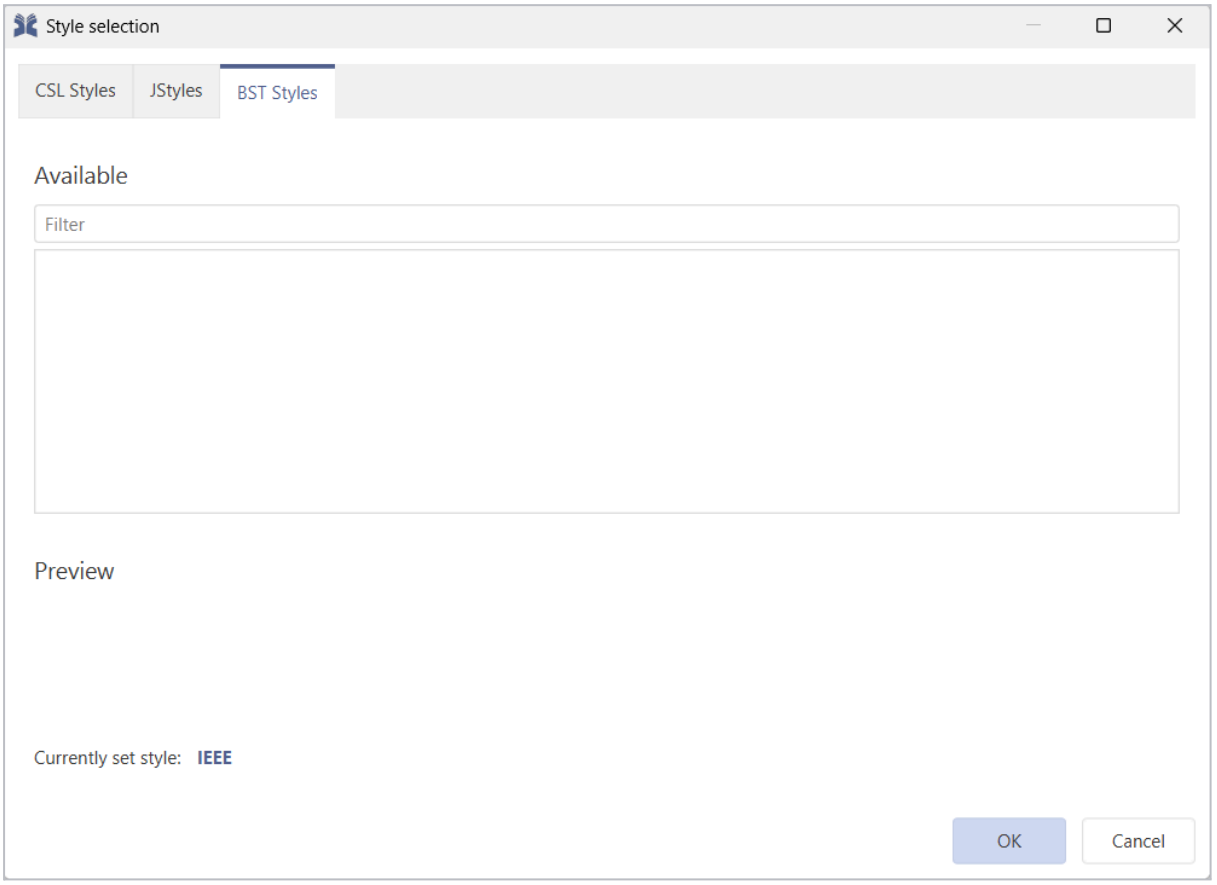
How BST Style Support is Implemented?

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, 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">
      <Label text="%Available" styleClass="sectionHeader"/>
      <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.

```
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.

A **BSTStyle.java** class needs to be created, similar to JStyle and CitationStyle, to manage variables, file state, and other related functionalities.

How to Improve Support for CSL Styles?

A.) Footnote-based Citation Support for CSL Styles

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.

The CSL processor will be modified to recognize and dynamically load custom styles. A validation mechanism will be implemented to ensure that imported styles are correctly parsed and stored.

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

How Citation Style Type Switching is Implemented?

LibreOffice currently supports reference marks for citations, but there is no automated process to update and apply citation styles directly within the document.

Identify Existing Reference Marks
LibreOffice's UNO API will be utilized to retrieve all reference marks within the document. These references will be extracted as **XTextRange** objects and structured for standardized processing.

The following method retrieves the bibliography range:

```
public Optional<XTextRange> getBibliographyRange(XTextDocument doc)
        throws NoDocumentException, WrappedTargetException {
    LOGGER.debug("Attempting to get bibliography range");
    Optional<XTextRange> range = UnoTextSection.getAnchor(doc, BIBLIOGRAPHY_SECTION_NAME);
    LOGGER.debug("Bibliography range found: {}", range.isPresent());
    return range;
}
```

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.

Apply the New Citation Style
A citation style selection interface will be introduced, allowing users to apply predefined formats from JabRef. This involves converting standardized references into the selected style.

Update Document with New Citations
The system will replace existing reference marks with the newly formatted citations, ensuring uniformity across the document.

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 ExistingReferenceMarks DO
        // Parse current reference marks into a unified format
        StandardizedEntry = parseReferenceMark(Reference)

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

        // Replace old references with new ones in the document
        updateReferenceMark(doc, Reference, ConvertedEntry)

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

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

How is Cross-Compatibility Implemented with other Reference Management Software?

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 integrated into OpenOffice, similar to the existing `JabRef_LibreOffice_Converter`, to facilitate a smooth transition for users.

Ensuring Bidirectional Compatibility

Importing citations from Zotero and Mendeley into JabRef will be tested to ensure correct interpretation. Tests will be written to confirm that LibreOffice documents containing citations from multiple tools function without formatting errors.

Timeline

	Start	**End**	**Tasks**
Community bonding period	May 08	June 01	Start coding for the project. Lot of time will be saved here since I already have a deep understanding of the codebase.
	June 02	June 04	Implement UI for BST Style tab
	June 05	June 07	Create the class BSTStylel.java and add the logic to make the UI work.
	June 08	June 12	Write tests and check after choosing the style, the current style is updated.
	June 13	June 17	Deeply analyze the implementation of numeric citation markers for JStyle.
	June 18	June 21	Add implementation like JStyle for CSL Style.
	June 22	June 24	Write tests to check the functionality.
	June 25	June 27	Create UI for importing custom CSL Styles.
	June 28	July 04	Provide implementation for processing imported custom CSL styles.
	July 05	July 11	Write tests to check the functionality.
Mid-term Evaluation	July 12	July 18	Evaluate the work done in phase-1 and make it ready for the first evaluation.

	July 18	July 21	Learn more about how to integrate LibreOffice with software using Java.
	July 22	July 25	Implement the functionality for identifying existing reference marks and parsing.
	July 25	July 30	Add the code for auto-updation of citations.
	Aug 31	Aug 04	Write tests to check the functionality.
	Aug 05	Aug 07	Modify the internal citation format.
	Aug 08	Aug 12	Implement converter for migrating citations.
	Aug 13	Aug 18	Integrate converter with OpenOffice.
	Aug 19	Aug 21	Ensure compatibility with Zotero and Mendely.
	Aug 22	Aug 24	Write tests to check the functionality.
	Aug 24	Aug 26	Refactor the code implemented and optimize.
	Aug 27	Sep 01	Write documentation and blog posts for the features.

Contributions so far

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

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

Overall Contribution

Total PRs	25
Total Issues Raised	5

Why me?

I've been actively contributing to JabRef since December 2024 and my contributions are the top contributions since I started contributing (after maintainers).

I've dedicated the last 1 month doing research on this project and have a clear mindset to complete this project.

With my experience and research, I am confident in contributing efficiently to JabRef's growth.

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 feature 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 🙂