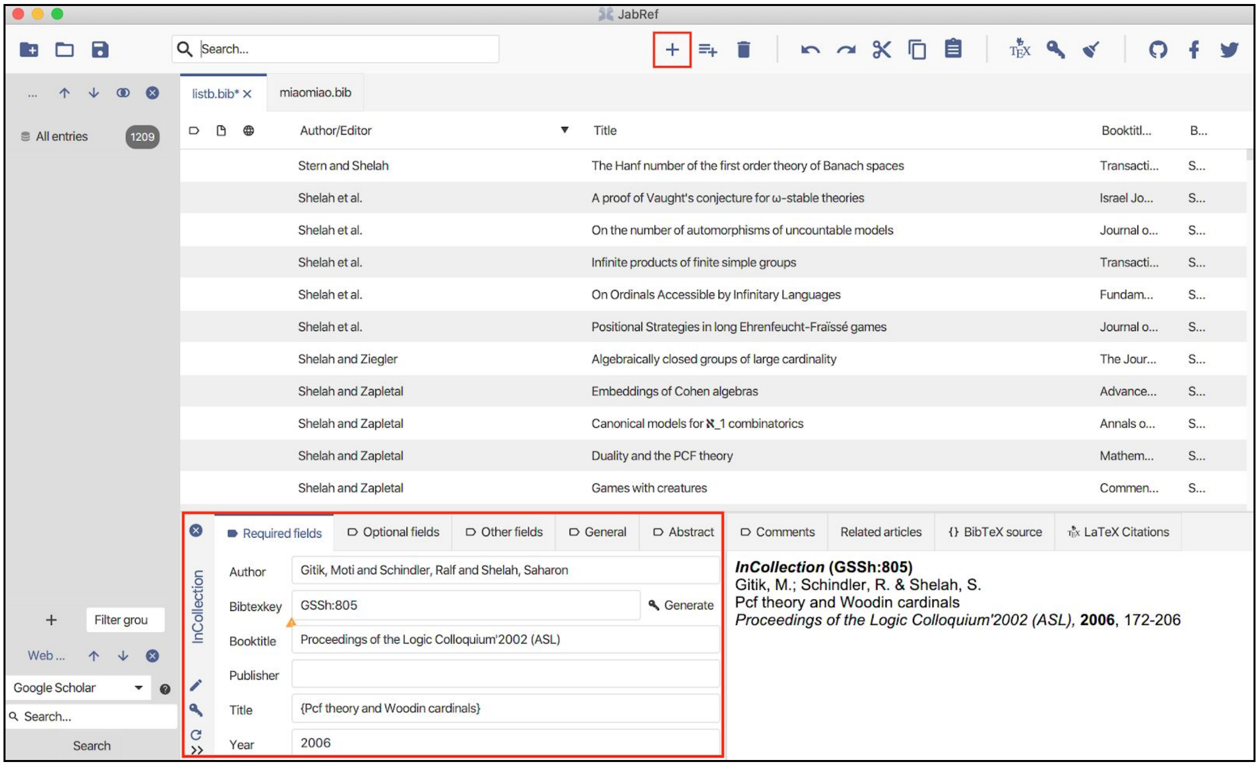
Feature: Add a new article

Add a new article (a single entry) using the “+” button.



Author/Editor	Title	Booktitl...	B...
Stern and Shelah	The Hanf number of the first order theory of Banach spaces	Transacti...	S...
Shelah et al.	A proof of Vaught's conjecture for ω-stable theories	Israel Jo...	S...
Shelah et al.	On the number of automorphisms of uncountable models	Journal o...	S...
Shelah et al.	Infinite products of finite simple groups	Transacti...	S...
Shelah et al.	On Ordinals Accessible by Infinitary Languages	Fundam...	S...
Shelah et al.	Positional Strategies in long Ehrenfeucht-Fraïssé games	Journal o...	S...
Shelah and Ziegler	Algebraically closed groups of large cardinality	The Jour...	S...
Shelah and Zapletal	Embeddings of Cohen algebras	Advance...	S...
Shelah and Zapletal	Canonical models for ℵ_1 combinatorics	Annals o...	S...
Shelah and Zapletal	Duality and the PCF theory	Mathem...	S...
Shelah and Zapletal	Games with creatures	Commen...	S...

Author: Gitik, Moti and Schindler, Ralf and Shelah, Saharon
Bibtexkey: GSSh:805
Booktitle: Proceedings of the Logic Colloquium'2002 (ASL)
Publisher:
Title: {Pcf theory and Woodin cardinals}
Year: 2006

InCollection **(GSSh:805)**
Gitik, M.; Schindler, R. & Shelah, S.
Pcf theory and Woodin cardinals
Proceedings of the Logic Colloquium'2002 (ASL), **2006**, 172-206

Figure (1) : Entry Editor feature

Reason: As a citation and reference management tool, JabRef helps users collect and organize resources in research. Adding a new entry with the type of article is very important as it keeps track of different research articles which the users are utilizing.

Components:

1. **JabRefFrame.java** `package org.jabref.gui;`
   This class initializes the main window of the JabRef application. Inside the createToolbar() method, the tool bar which includes the plus sign button gets created. Specifically, the createIconButton() method of ActionFactory.java takes in this NEW_ARTCILE action and corresponding command, which in this case is a NewEntryAction.

```
HBox rightSide = new HBox(
        factory.createIconButton(StandardActions.NEW_ARTICLE, new
NewEntryAction(this, StandardEntryType.Article, dialogService,
Globals.prefs, stateManager)), ....)
```

2. **NewEntryAction.java** (package org.jabref.gui.importer)
   When users click the plus sign, the execute() method of NewEntryAction.java gets called subsequently. There can be many types of entries such as articles, books, journals, etc. With the present “article” type, the current JabRefFrame gets the current BasePanel, and executes the function of inserting new bib entry.

```
@Override
public void execute() {
   if (jabRefFrame.getBasePanelCount() <= 0) {
       LOGGER.error("Action 'New entry' must be disabled when no
database is open.");
       return;
   }
   if (type.isPresent()) {
      jabRefFrame.getCurrentBasePanel().insertEntry(new
BibEntry(type.get()));
   } else {
       EntryTypeView typeChoiceDialog = new
EntryTypeView(jabRefFrame.getCurrentBasePanel(), dialogService,
preferences);
       EntryType selectedType =
typeChoiceDialog.showAndWait().orElse(null);
       if (selectedType == null) {
           return;
       }
       trackNewEntry(selectedType);
      jabRefFrame.getCurrentBasePanel().insertEntry(new
BibEntry(selectedType));
   }
}
```

3. **BasePanel.java** (package org.jabref.gui)
   The following methods get called immediately.

```
public void insertEntry(final BibEntry bibEntry) {
   if (bibEntry != null) {
       insertEntries(Collections.singletonList(bibEntry));
   }
}
```

```
public void insertEntries(final List<BibEntry> entries) {
    if (!entries.isEmpty()) {
        try {
            bibDatabaseContext.getDatabase().insertEntries(entries);

            // Set owner and timestamp
            for (BibEntry entry : entries) {
                UpdateField.setAutomaticFields(entry, true, true,
Globals.prefs.getUpdateFieldPreferences());
            }
            // Create an UndoableInsertEntries object.
            getUndoManager().addEdit(new
UndoableInsertEntries(bibDatabaseContext.getDatabase(), entries));

            markBaseChanged(); // The database just changed.
            if
(Globals.prefs.getBoolean(JabRefPreferences.AUTO_OPEN_FORM)) {
                showAndEdit(entries.get(0));
            }
            clearAndSelect(entries.get(0));
        } catch (KeyCollisionException ex) {
            LOGGER.info("Collision for bibtex key" + ex.getId(), ex);
        }
    }
}
```

4. **BibDatabaseContext.java** (`package org.jabref.model.database`)
   On the database side, this new entry is also inserted into the current database with the help of BibDatabaseContext, which gets the database object and has it inserting the new entry.

5. **BibDatabase.java** (`package org.jabref.model.database`)
   It is noticed that in this code block, eventbus's post method gets fired and posts a new event of EntriesAddedEvent. MainTable, on the other side, subscribes to the

eventbus. Every time we add a new entry into the bib database, MainTable's listen method will be invoked.

```
public synchronized void insertEntries(List<BibEntry> newEntries,
EntriesEventSource eventSource) throws KeyCollisionException {
   Objects.requireNonNull(newEntries);
   for (BibEntry entry : newEntries) {
       String id = entry.getId();
       if (containsEntryWithId(id)) {
           throw new KeyCollisionException("ID is already in use,
please choose another", id);
       }

       internalIDs.add(id);
       entry.registerListener(this);
   }
   if (newEntries.isEmpty()) {
       eventBus.post(new EntriesAddedEvent(newEntries, eventSource));
   } else {
       eventBus.post(new EntriesAddedEvent(newEntries,
newEntries.get(0), eventSource));
   }
   entries.addAll(newEntries);
}
```

6. **Maintable.java** (`package org.jabref.gui.maintable`)

```
@Subscribe
public void listen(EntriesAddedEvent event) {
   DefaultTaskExecutor.runInJavaFXThread(() ->
clearAndSelect(event.getFirstEntry()));
}
```

This clears the highlight of the currently selected row and highlights the currently editing row for the new entry.

```
public void clearAndSelect(BibEntry bibEntry) {
   findEntry(bibEntry).ifPresent(entry -> {
       getSelectionModel().clearSelection();
       getSelectionModel().select(entry);
       scrollTo(entry);
   });
}
```

6. **EntryEditorTab.java** (`package org.jabref.gui.entryeditor)`
The entry editor tab now becomes the focus.

```
/**
* The tab just got the focus. Override this method if you want to
perform a special action on focus (like selecting
* the first field in the editor)
*/
protected void handleFocus() {
   // Do nothing by default
}
```