Feature: MainTable

Each tab(lies on the left side panel named “Groups”) is a BasePanel. BasePanel has a search bar at the top, a MainTable at the center and a PreviewPanel or an EntryEditor at the bottom.

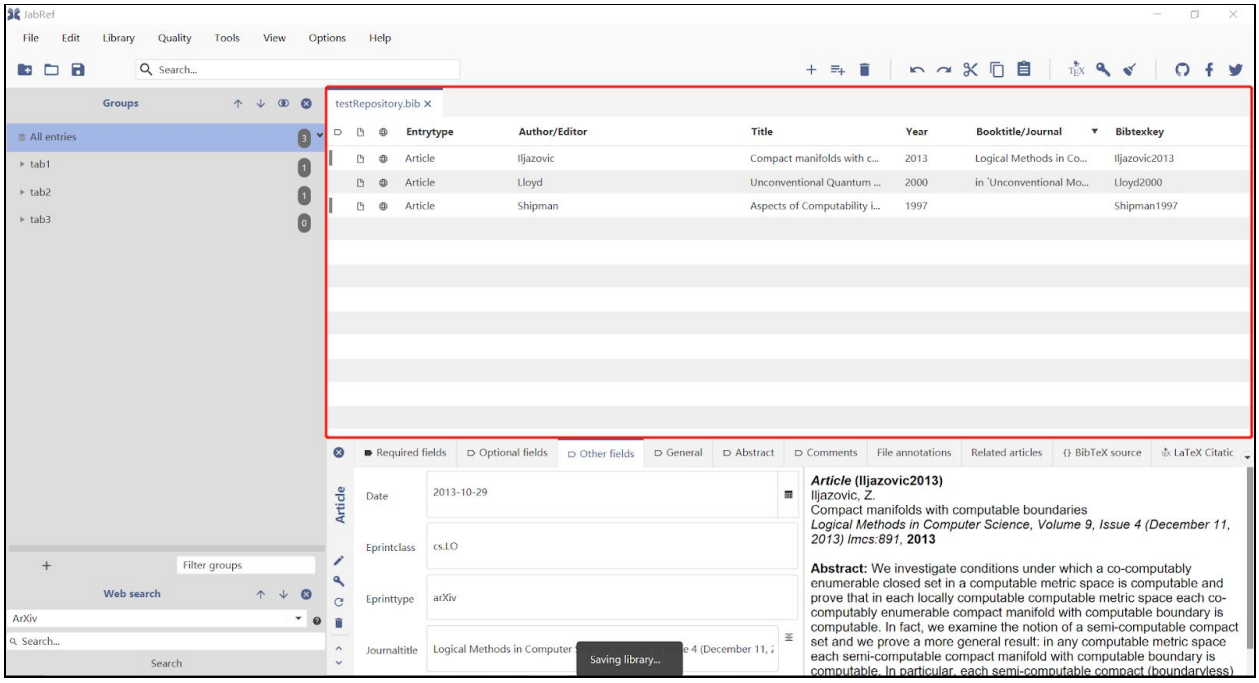


Figure (1) : MainTable feature

Reason: The MainTable occupies the main part of the user interface. Since JabRef is a bibliography reference manager, some basic essential aspects of the reference will be shown on the MainTable. Hence, MainTable is an indispensable feature in the JabRef project. Therefore, our team wants to further explore this feature at a higher level, to identify how different components are assembled.

Source Code location: package org.jabref.gui.maintable. Locations of other related codes could be found in the “Components” part.

Components:

1. **BasePanel**

BasePanel extends StackPane, which is a layout in JavaFX. The BasePanel class could be found in package org.jabref.gui. BasePanel sets up a set of actions, such as

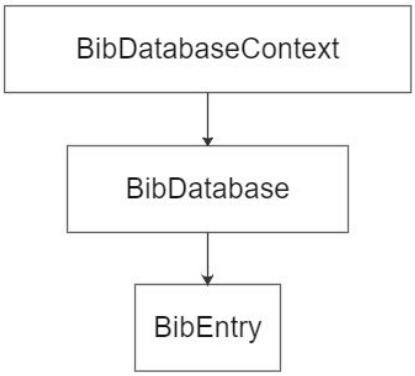
undo, edit, save, cut, delete, etc. Whenever you want to create a new MainTable object, a BasePanel object must be included.

2. **Database**

Every entry is stored in the corresponding database. In the MainTable.java (org.jabref.gui.maintable), we can see a BibDatabaseContext called “database”.

```
private final BibDatabaseContext database;
```

BibDatabaseContext represents everything related to a Bib file. It contains a BibDatabase class, in which data entries are stored.



The plot above shows the hierarchy of the database classes. Related classes can be found in package org.jabref.model.database. Classes related to entries could be found in package org.jabref.model.entry. Besides, Bibtex database fields could be found in org.jabref.model.entry.field.StandardField.java. Fields include author, book title, date, etc.

3. **ViewModelTableRowFactory**

Used for binding click events, like using a mouse to drag an entry. Codes below could be found in org.jabref.gui.MainTable.maintable.java. To see how those specific functions are defined, check package org.jabref.gui.util.ViewModelTableRowFactory.java.

```
new ViewModelTableRowFactory<BibEntryTableViewModel>()
                .withOnMouseClickedEvent((entry, event) -> {
                                                                    if
(event.getClickCount() == 2) {

panel.showAndEdit(entry.getEntry());
```

```
                                                                    }
                                                                })
                .withContextMenu(entry ->
RightClickMenu.create(entry, keyBindingRepository, panel,
frame.getDialogService()))
                .setOnDragDetected(this::handleOnDragDetected)
                .setOnDragDropped(this::handleOnDragDropped)
                .setOnDragOver(this::handleOnDragOver)
                .setOnDragExited(this::handleOnDragExited)
                .setOnMouseDragEntered(this::handleOnDragEntered)
                .install(this);
```

4. RightClickMenu

Any right click on the MainTable is handled by the RightClickMenu (package org.jabref.gui.maintable).

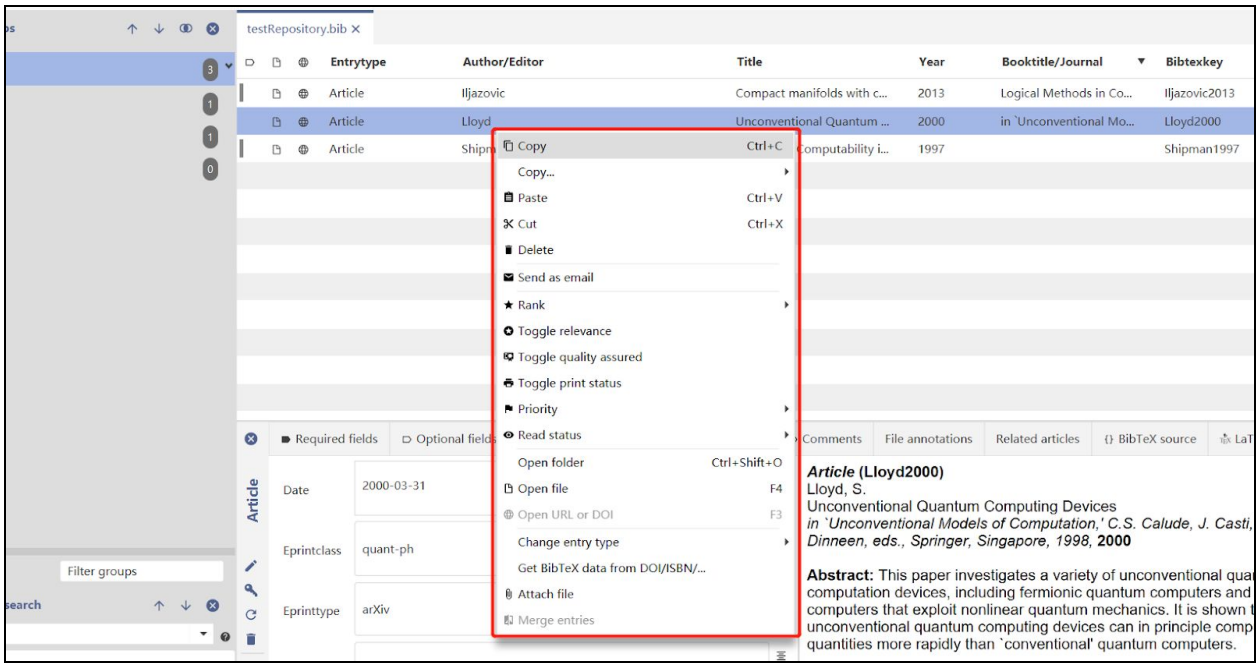


Figure (2) : RightClickMenu in MainTable feature

Functions in ActionFactory.java are called when defining functions in RightClickMenu. Other related classes could be found in package org.jabref.gui.actions. Code below shows how different operations are added to the context of the right click menu.

```
ContextMenu contextMenu = new ContextMenu();
```

```
ActionFactory factory = new ActionFactory(keyBindingRepository);

contextMenu.getItems().add(factory.createMenuItem(StandardActions.COP
Y, new OldCommandWrapper(Actions.COPY, panel)));
contextMenu.getItems().add(createCopySubMenu(panel, factory,
dialogService));
contextMenu.getItems().add(factory.createMenuItem(StandardActions.PAS
TE, new OldCommandWrapper(Actions.PASTE, panel)));
contextMenu.getItems().add(factory.createMenuItem(StandardActions.CUT
, new OldCommandWrapper(Actions.CUT, panel)));
contextMenu.getItems().add(factory.createMenuItem(StandardActions.DEL
ETE, new OldCommandWrapper(Actions.DELETE, panel)));
```

5. **Maintable.css**

Defines style and how elements should be played. It's in the package org.jabref.gui.maintable.

6. **importHandler**

In the MainTable.java (org.jabref.gui.maintable), we can see a ImportHandler called "importHandler ".

```
private final ImportHandler importHandler;
```

Below is the specific piece of code.

```
importHandler = new ImportHandler(
       frame.getDialogService(), database, externalFileTypes,
       Globals.prefs.getFilePreferences(),
       Globals.prefs.getImportFormatPreferences(),
       Globals.prefs.getUpdateFieldPreferences(),
       Globals.getFileUpdateMonitor(),
       undoManager,
       Globals.stateManager);
```

This object initializes preferences, such as format, file preferences, languages, themes, etc. Related classes could be found in org.jabref.preferences.JabRefPreferences.java**.**