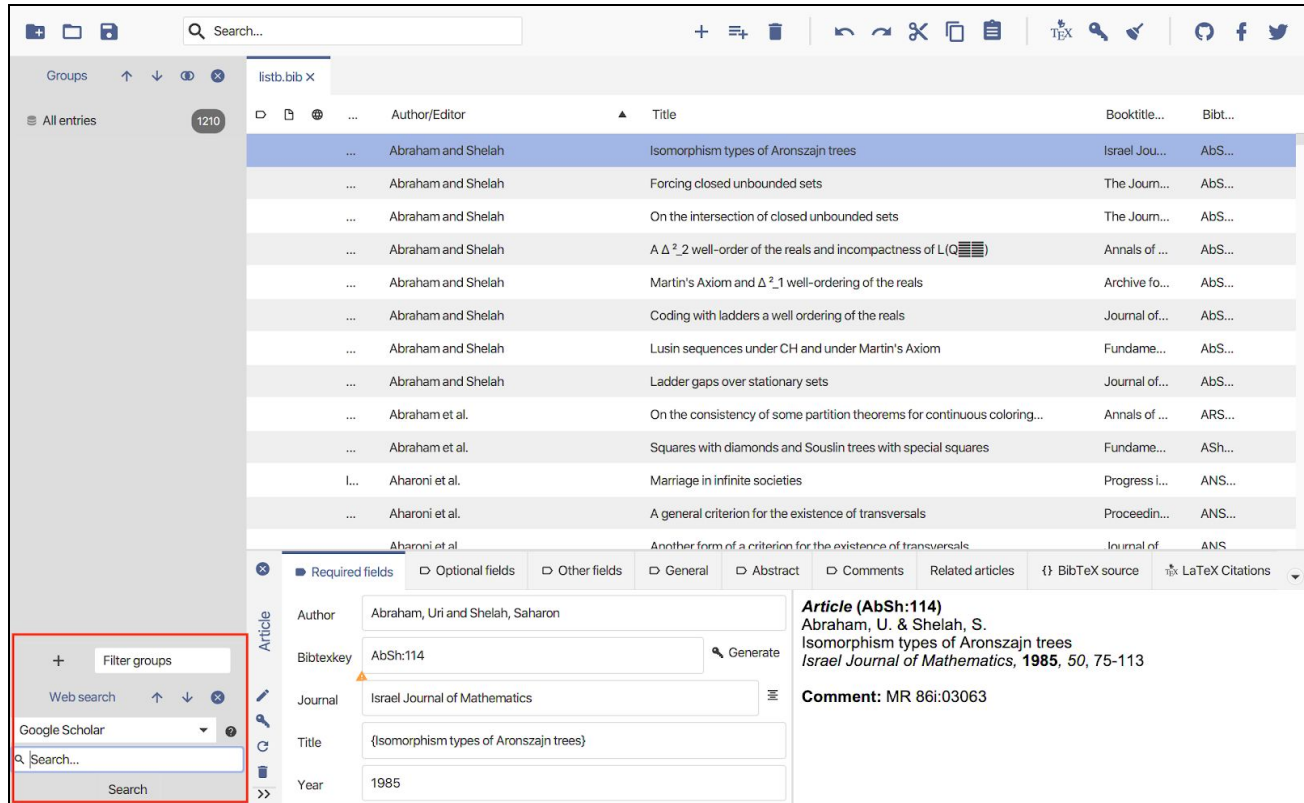


## SWE 265P Reverse Engineering and Modeling

Project: JabRef - an open-source, cross-platform citation and reference management tool.

### Feature 1. Web Search



*Figure (1) : User Interface highlighting web search*

We ran the project and had some interest in a very convenient function called Web Searching, which allows users to search across many online scientific catalogs such as CiteSeer, CrossRef, Google Scholar, etc. The user interface of this feature is shown in figure 1. We tried this function and added two entries to our own repository. To start finding out how this feature is built, we checked the UML plots first. It is a huge UML plot and hence, spent a long time on it. Finally, we noticed that there is a class called SidePaneComponent. Since the Web Searching function lies on the side panel of the main interface, we thought this class might have some relation with the Web Search function. Then, we found a class called WebSearchPane that extends the abstract class, SidePaneComponent, which led us to our starting point. During our search process, we realized that this function could be divided into two parts: fetching data from web & importing data to the local database. The table below contains the details.

**Team : C#-er than you**  
**Assignment 1**

ID	Folder	File	Method	Why	Next
1	org.jabref.logic.importer	SidePaneComponent	Abstract class	Entry point	2
2	org.jabref.gui.importer.fetcher	WebSearchPane	None	Extends SidePaneComponent	3
3	org.jabref.gui.importer.fetcher	WebSearchPaneViewModel	search	The search button invokes this function	4
4(1)	org.jabref.logic.importer	SearchBasedFetcher	performSearch	Thought the feature processed search queries but wasn't fruitful	4(2)
4(2)	org.jabref.gui.importer.fetcher	WebSearchPaneViewModel	ImportEntriesDialog	If we run the project, a new dialog box pops after we click search. Hence, this looks about right	5
5	org.jabref.logic.importer	WebFetchers	getSearchBasedFetchers	The page gets data from the web in here	6
6	org.jabref.gui.importer	ImportEntriesDialog	ImportEntriesDialog	Tried to find the import function	7
7	org.jabref.gui.importer	ImportEntriesViewModel	importEntries	"import" button invokes this function to add the item into local	Done

*Table (1) : Web Search feature - Search template*

Related code snippets for Web Search feature:

```
public void importEntries(List<BibEntry> entriesToImport) {
    // Check if we are supposed to warn about duplicates.
    // If so, then see if there are duplicates, and warn if yes.
    if (preferences.shouldWarnAboutDuplicatesForImport()) {
        BackgroundTask.wrap(() -> entriesToImport.stream()
            .anyMatch(this::hasDuplicate)).onSuccess(duplicateFound -> {
            if (duplicateFound) {
                boolean continueImport = dialogService.showConfirmationDialogWithOptOutAndWait(Localization.lang
                    ( key: "Duplicates found"),
                    Localization.lang( key: "There are possible duplicates (marked with an icon) that haven't " +
                        "been resolved. Continue?"),
                    Localization.lang( key: "Continue with import"),
                    Localization.lang( key: "Cancel import"),
                    Localization.lang( key: "Disable this confirmation dialog"),
                    optOut -> preferences.setShouldWarnAboutDuplicatesForImport(!optOut));

                if (!continueImport) {
                    dialogService.notify(Localization.lang( key: "Import canceled"));
                } else {
                    buildImportHandlerThenImportEntries(entriesToImport);
                }
            } else {
                buildImportHandlerThenImportEntries(entriesToImport);
            }
        }).executeWith(Globals.TASK_EXECUTOR);
    } else {
        buildImportHandlerThenImportEntries(entriesToImport);
    }
}
```

*Code snippet (1) : ImportEntriesViewModel.java [Lines 75 - 104]*

## Feature 2. Open JabRef's Github/Facebook/Twitter Page

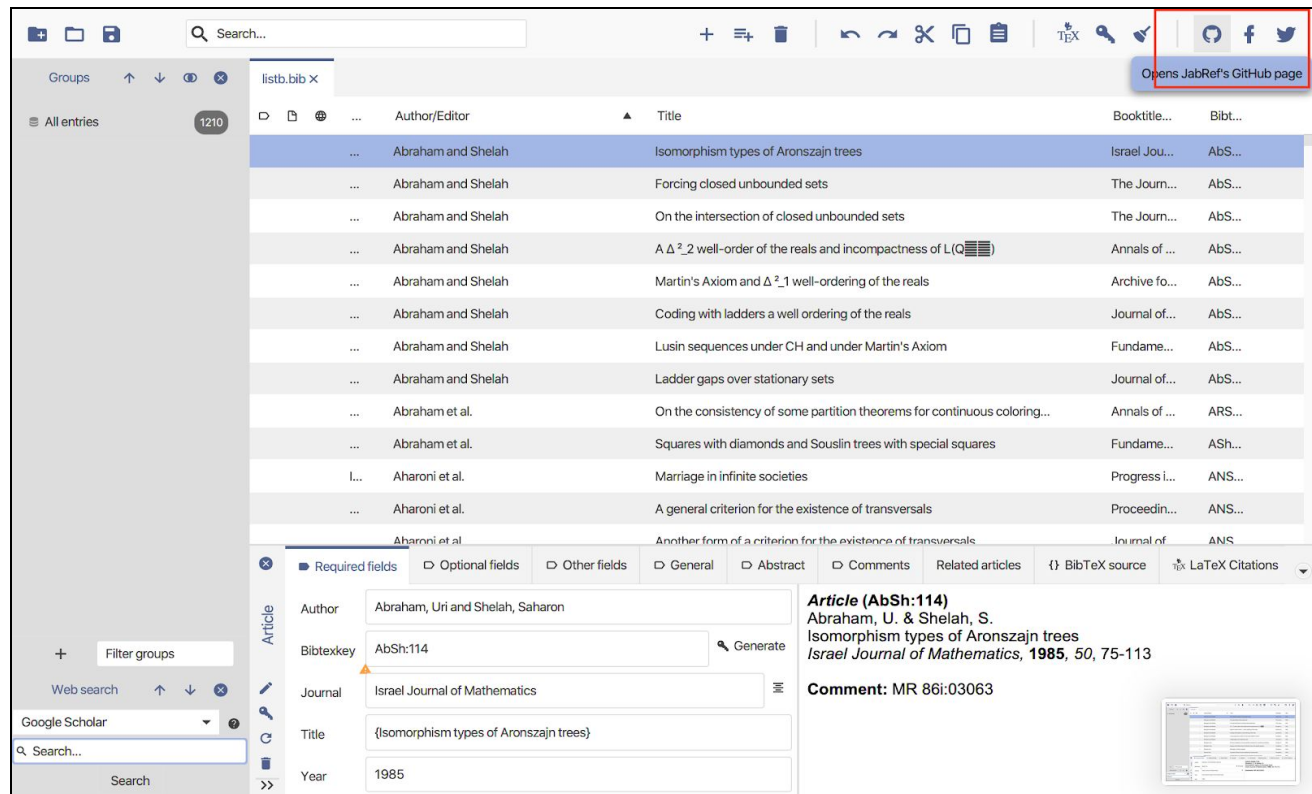


Figure (2) : User Interface highlighting social media handle

This feature is straight-forward. It opens a corresponding URL when users click one of the social media icons located in the up-right corner of the application interface. We began exploring our UML plots, but this strategy was not fruitful. After giving some thought, we realized that "Facebook" is a special keyword, so we looked for this keyword using the "Find in the path" function. The result showed that there is a variable called FACEBOOK in the IconTheme.java file. So we used this as a starting point to find more clues. The whole feature could be divided into two parts: creating the icon button and opening the URL. Operations related to creating the icon button are located in the ActionFactory.java and the operations related to opening the URL are in the OpenBrowserAction.java. The table below contains the details.

ID	Folder	File	Method	Why	Next
1	org.jabref.gui.icons	IconTheme	None	'Facebook' is a special keyword, we use 'Find in path' to search for 'Facebook'	1
2	org.jabref.gui.actions	StandardActions	OPEN_FACEBOOK variable	Found usage following the previous step	3
3	org.jabref.gui	JabRefFrame	HBox	Found usage following the previous step	4
4(1)	org.jabref.gui.actions	ActionFactory	createIconButton	This function creates the iconButton	Done
4(2)	org.jabref.gui.editor	OpenBrowserAction	OpenBrowserAction	This function opens the URL	Done

*Table (2) : Social media handle feature - Search template*

Related code snippets for Social Media handle feature:

```
public Button createIconButton(Action action, Command command) {  
    Button button = ActionUtils.createButton(new JabRefAction(action, command, keyBindingRepository,  
        Sources.FromButton), ActionUtils.ActionTextBehavior.HIDE);  
  
    button.getStyleClass().setAll("icon-button");  
  
    // For some reason the graphic is not set correctly, so let's fix this  
    button.graphicProperty().unbind();  
    action.getIcon().ifPresent(icon -> button.setGraphic(icon.getGraphicNode()));  
  
    return button;  
}
```

*Code Snippet (2) : ActionFactory.java [Lines 129 - 140]*

```
public OpenBrowserAction(String urlToOpen) {  
    this.urlToOpen = urlToOpen;  
}
```

*Code Snippet (3) : OpenBrowserAction.java [Lines 10 - 12]*