| Symbol | Meaning |
| --- | --- |
| !! | Must have |
| ! | Should have |
| $ | Could have |
| + | Nice to have |
| X | Conflict |

# Core functions

* Delete files !!
* Dark mode !
* Should be possible to use with only one reviewer !!
* Should be possible to use with multiple reviewers !! # Uploading
* Batch upload !!
  + From local folder
  + Should be able to pull from APIs $
* Support uploading PDFs initially !!
* Should be able to use other formats $
  + EndNote XML, PubMed, RIS, Zotero

# Categories

Be able to accept/reject/skip files

Statuses: - Yes !! - No !! - Conflict !! - Goes into queue for later - Later !! - Goes into backlog

# Search filtering

* Search the index of the files !
  + Filter by: author, keyword, publication date (year range)
  + Multiple filters can be used at a time
  + These should be buttons - not a query language

# Exporting

* At any stage
* What was accepted, rejected, reviewers, papers used !!
  + Export the key data (E.G statistics) from each paper at the end of the review $
* Pick output format $
* Pick referencing style $
* Summarise with LLM +

# Collaboration

* Reviewers can’t see what the other reviewers have voted until they have completed the same stage !!
* Live collaboration +

## Roles

* Each license has an owner
  + One license may have many projects
  + Each project has some owners and some reviewers
* Roles are not mutually exclusive

### Project owner !!

* See names of reviewers
* Manage reviewers
* There must be at least one owner

### Reviewers !!

#### Core reviewer !

* Their votes are required
* In team reviews, there must be at least

#### Conflict resolver !

* Solves conflicts
* Votes are not required for anything but conflicts

#### Spectator +

* Can view everything, but cannot do anything

# Highlighting

* Set include-exclude criteria !
* Configure per stage +
* Use word association ++

# Dashboard !

## Main dashboard !

* Overview of statuses of all files: !
  + Finished
  + One vote
  + Conflict
  + No votes
* Statistics +
  + Acceptance rate
  + Rejection rate
  + This would help the managers assess the team’s progress through the workload
  + Break down by reviewer if the menu is clicked? +
* Embed productivity management software ++ ## Owner dashboard +

# Notifications

* Emails when: !
  + You’re added to a project
  + When you are assigned work
  + There is a conflict you are responsible for
* Team communication software integration: +
  + MS Teams
  + Google Workspaces

# Paper queues !!

* To be reviewed !!
* Reviewed with categories: !
  + Accepted
  + Rejected
* Conflicted !!
  + Separate queue
  + Should not interrupt main workflow
* The status of the file should be visible !
  + The voting status
  + The stage it’s in
  + Whether it’s in conflict
  + Should be able to view files grouped by status

## Conflict resolution !!

* Conflict resolvers are notified !
* They must vote in order for the paper to exit the conflict queue !!
* Conflict resolution strategy should be configurable at the project level:
  + Intermediary strategy
  + Re-voting by core reviewers

# Assign workload +

* Core viewers assign workloads to normal reviewers
* Core reviewer effectively becomes a management role
* Work out the details of this

# PDF Viewer !

* Zoom
* Comment
* Heading outline
* Mini-map +
* Search text
* Highlight relevant words (inclusion/exclusion)
  + Highlighting criteria menu
* Split into relevant sections (title, abstract, full text)
* Vote buttons (accept, reject, skip)

## Security

* Files should be encrypted when they are uploaded to the database !
* Sign in with: !!
  + Google
  + SSO

# AI assistance +

* Search papers based on prompt
* Sort by what the AI views as relevant
* The AI votes and the reviewer must approve/deny the vote
  + The AI will give a reason for its vote # Questions
* When one person finishes a stage, should they have to wait for the other reviewers to finish the stage before progressing?
  + Pro: prevents redundant work in later stages
  + Con: blocks reviewers
  + Suggestion: let users review files that they have already voted yes on, but if the other reviewers vote no, the file gets dragged back to the previous stage
  + Pending state
* Should the minimum number of core reviewer votes be configurable or automatic?
  + Useful for single-reviewer projects (I.E personal use)
* What are the requirements for auditing the review process?
  + A searchable timeline of all actions that occurred?