

Project outline

Backend and Web Frontend

- users
 - login
 - register
 - teams
 - join teams
 - invite to teams
 - user roles
- book Project
 - create
 - upload source (book to be translated)
 - add team or individual users to book (will get rights to download and send pull requests) (pull request should be called review request)
 - approve pull requests (admin only)
 - decline pull requests (admin only)
 - download master branch as book file
 - create release for print
 - see differences between master branch and current release
 - view book in its entirety

GUI Client

- login
- view list of book projects this user is a part of
- download latest version of a book project
- continue translation
- side by side translation (viewing original next to new translation)
- user can only edit one paragraph at the time
- enter button doesn't work in a paragraph
- save and/or push changes to admin for review
- force download of updates so translators always work on the latest approved copy
- if source book exist paragraphs are locked to the paragraphs of the source. No new ones can be created.
- creation of paragraphs and chapters if no source book exists

How a book project works?

Each book project is stored individually with or without a reference to the book its translated from with ISBN number and other important info.

A book is is stored as multiple files in a file system. With a json manifest file which structures the book from all individual files. Each file is a paragraph of the book and stored individually in folders called chapters. So each Chapter folder contains all the paragraphs that make up all the content of that chapter. ([TO DISCUSS] should each chapter contain its own manifest which includes any data for that chapter like title or description etc. and the main manifest builds from each chapter manifest?). The reason for one file per paragraph is to give the possibility to re-use paragraphs in other book projects and to be to link that paragraph but not edit it as it belongs to another project. This way books using paragraphs from other books will always be up to date with the latest translation.

Each book project is its own git project. By using the functionality of git users can work individually and suggest any changes to the admin who can pick the changes and care for the entirety of the book. Contributors push changes and the admin merges any changes that are ok. Git can then allow comparisons of different branches and files to make it easy to get an overview of exact changes. The git part should be simplified to a level that anyone without git experience can use it in this context.

Each book project has at least one admin and can have multiple contributors.