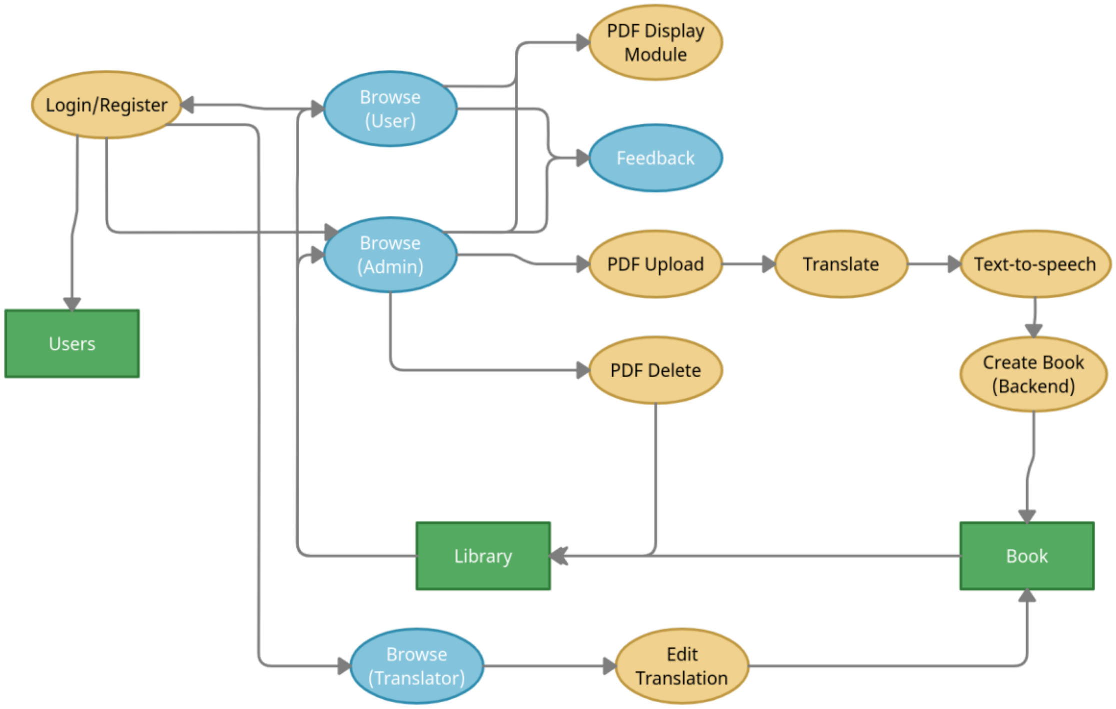
Team 23, eBook Reader, Abhijeeth Singam, Ainesh Sannidhi, Aryan Jain, Palash Sharma

Design Overview

Architectural design

The below diagram is a representation of how the different users interact with the application. Each blue oval represents a page the user can visit. Each golden oval represents a module of code that can be triggered via user interaction. Each green rectangle represents an entity stored in the backend of the application.

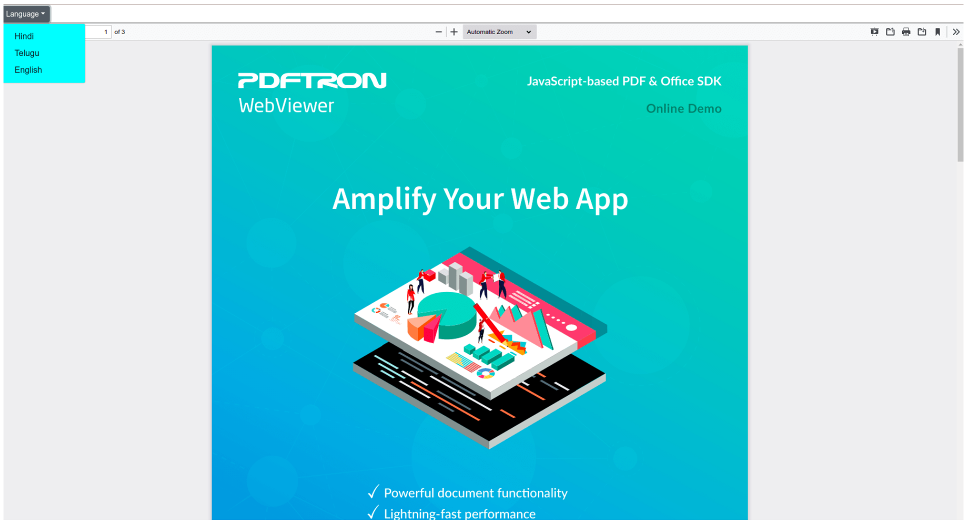


System interfaces

User Interface

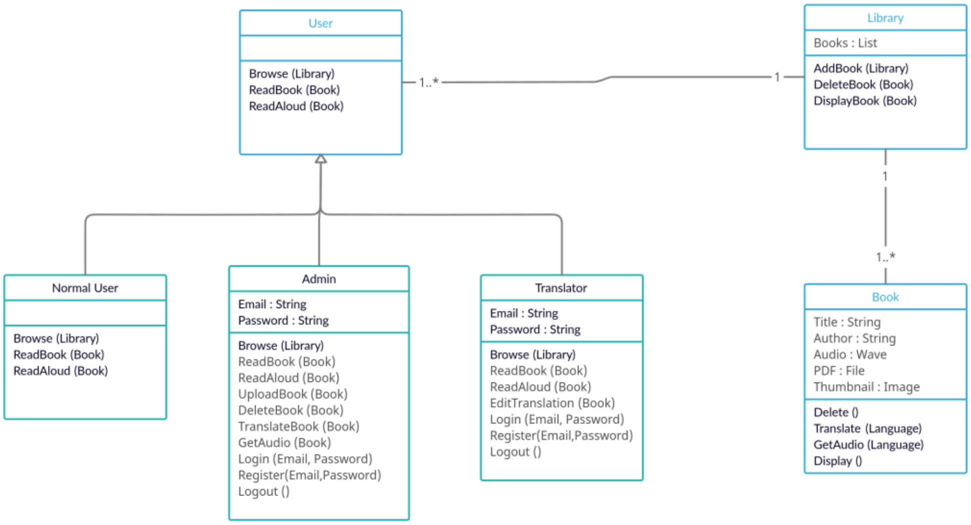
Users will arrive at the browse page which display a paginated list of all uploaded books along with a search bar for them to go through the list of books present in the library. This page will also contain details like the languages the book is available in and upon selecting a book the user will be taken to a PDF viewing page to read the selected book. From the PDF viewing page, users can select the language they wish to read the book in, and can also choose to have the book read aloud. Users may also select from a variety of options to change the view of the PDF, such as zoom in and zoom out.

Admins, once signed in, will have the additional functionality of uploading books, having the books translated and deleting previously uploaded books. Translators on the other hand will be able to modify the translated versions of the books to improve the translation quality.


Language
Hindi
Telugu
English
1 of 3
Automatic Zoom
PDFTRON
WebViewer
JavaScript-based PDF & Office SDK
Online Demo
Amplify Your Web App
✓ Powerful document functionality
✓ Lightning-fast performance

PDF Display Page

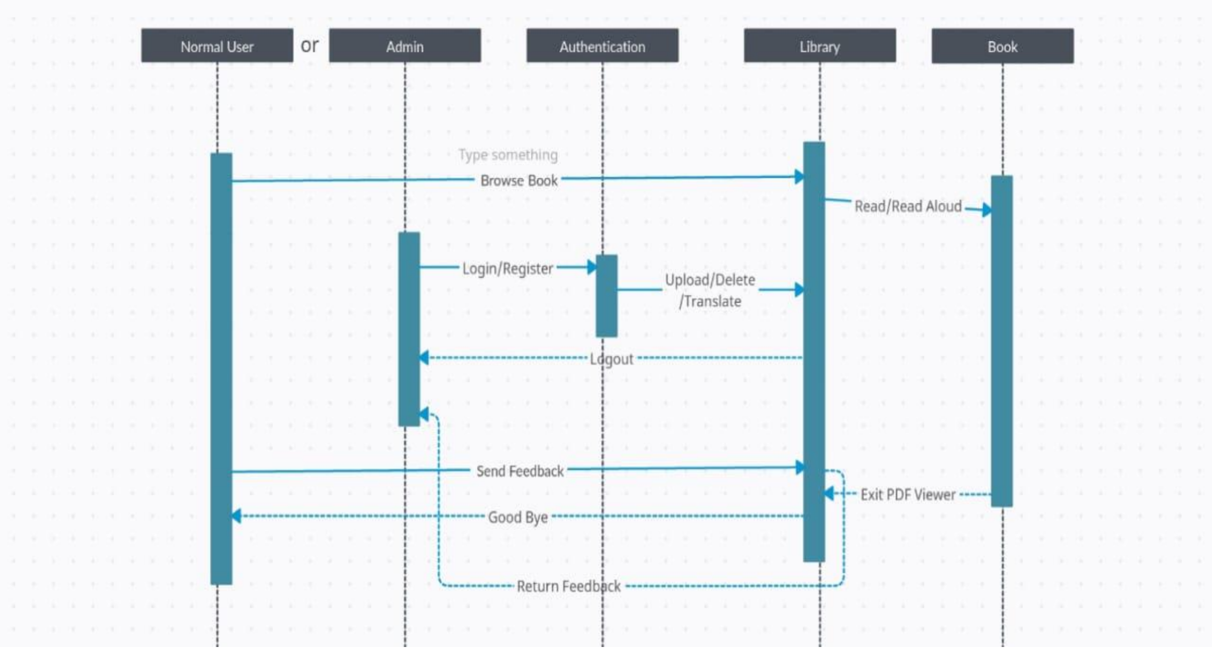
Model



Normal User	Class state Class behavior • Read Book • Read Aloud • Browse • Feedback
Admin	Class state • Email • Password Class behavior • Browse • Read Book • Read Aloud

	• Upload Book • Delete Book • Translate Book • Get Audio • Login • Register • Logout
Translator	Class state • Email • Password Class behavior • Browse • Read Book • Read Aloud • Edit Translation • Login • Register • Logout
Book	Class state • Title • Author • PDF • Audio • Thumbnail Class behavior • Translate • Delete • Get Audio • Display
Library	Class state • Books Class behavior • Add Book • Delete Book • Display Book

Sequence Diagram(s)



Design Rationale

- Project goal shifted from empowering the user to be able to translate and convert books to audio to instead providing a platform where users can read books, similar to audible.
- Use a direct ratio for font sizes between the original and translated versions
  - Considered Alternatives:
    - Change the font size of every single text block according to how much space each text block has. Rejected as it wouldn't produce a pleasant reading experience.
- Use materialUI components wherever possible in the frontend.
- Initial idea included the option for normal users to upload books and get it translated. But we later anticipated that allowing this would make the storage and API systems go haywire, which is why we differentiated normal users with admin users.
- Initial plan for text-to-speech was to perform text-to-speech translation as the user requested the text-to-speech to be made. Due to changes in the main goal and structure of the application, it was then changed to be done as soon as a book is uploaded and the audio files are stored in the backend along with the book.
- Initially had an overly minimalist-oriented wireframe design which was rejected since it was not to the taste of the general public and was not up to the standard of modern websites. Later shifted to more mainstream design ideas using basic materialUI components