P2Pで実現する本棚シェアリングWebアプリケーション

金沢工業大学 工学部 情報工学科
原田武長

研究背景・現状の問題点

個人や研究室など小規模な本棚向けの蔵書管理システムはあまりなく、本の貸し借りをしづらい状況がある。

口頭で本の貸し借りをする場合、相手がどんな本を持っているか毎回訊かなければならず、貸し借りの成立までの壁が高い。

既存のシステムは導入にあたりWebサーバー、データベースサーバー等を設置する必要があり、メンテナンス・維持をできる人が必要である。

研究目的

・本棚を見える化 することで本の貸し借りを
より身近にする
・蔵書管理システムの導入をより手軽にする

対象者

・個人/研究室等中小規模の本棚を持つ人
・これから蔵書管理システムを導入する人
・スマートフォンまたは PC のブラウザを利用できる人

解決案

個人や研究室の本棚をデータ化し、ブラウザ上で本の検索・貸し借り記録ができるWebアプリケーションを提案する。提案するWebアプリケーションでは、ブラウザ上で動作するP2Pライブラリであるjs-ipfs、およびIPFS内のコンテンツとしてデータベースを読み書きするOrbitDBを利用する。ブラウザ上で直接DBを操作することでバックエンドAPIを不要とし、システム導入を容易にする。

蔵書管理システムで扱うすべてのデータを、P2Pネットワーク内に保管することで非中央集権でサーバーが不要なシステムを実現する。これにより回線速度、可用性、メンテナンス等に期待できない環境下でも 蔵書管理システムの導入を可能にする。

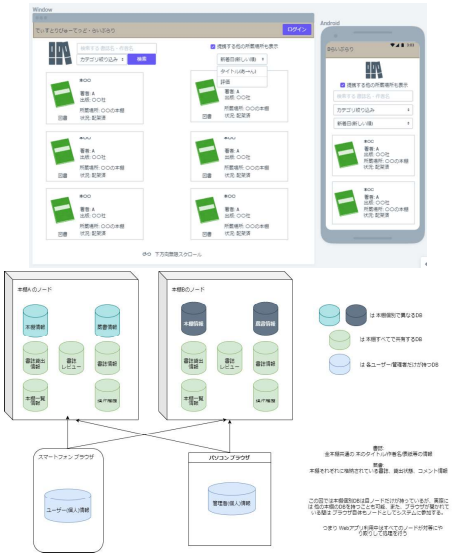
開発方法

開発手順

STEP1: サイトページ一覧のコンセプトデザインを作成
(蔵書一覧/本貸出返却/ユーザー/管理者 ページ等)
STEP2: サイトページを見ながら必要なDB群を書き出し
STEP3: DBに格納するデータの型/モデルをそれぞれ定義
STEP4: DBを操作するDAO(DatabaseAccessObject)群をそれぞれ作成
STEP5: 作成したDAOの動作検証
STEP6: DAOを利用するブラウザ上の画面を作成
STEP7: 動作テストを行いバグを修正する
STEP8: DAOを用いたOrbitDB操作にアクセス制限を実装(不正入力防止)

開発環境

Webアプリケーションフレームワーク: Nuxt.js(SPA)
開発言語: TypeScript
UIフレームワーク: Bulma
テストフレームワーク:Jest



利用方法

導入の流れ:

1. P2Pネットワークに配信するノードにできるコンピュータを最低1台、 本棚の管理者が用意する
2. 本研究で開発する蔵書管理Webアプリにアクセスし、新しい本棚とその管理者情報を初期化する
3. 初期化するとサイトを維持するために必要なDBの P2Pハッシュアドレスが表示される。本研究で開発する蔵書管理システム、および指定されたDBのハッシュアドレスを 配信ノードに追加し、維持する。
4. 任意のスマートフォン/コンピュータから P2Pノードでホストするサイトにアクセスする

ユーザーがサイト内で可能な操作:

複数の管理者の本棚の閲覧　本棚の管理者から本を借りる/返却する 記録を残す
(スマートフォンで書誌のISBNコードを読むことでDBに貸出/返却記録を保管する)

管理者がサイト内で可能な操作:

複数の管理者間で共通となる書誌情報DBの追加/編集
(ISBNコードを読み取ることでOpenBD等の書誌情報APIから書誌情報を取得する)
本棚内の本一覧となる、蔵書情報DBの追加/編集 貸出/返却 記録の確認

参考文献

書籍販売サイトと連携した研究室規模の蔵書管理システム-DB設計と基本機能-

ソーシャル図書室「bookHub」：コミュニティ指向の蔵書管理・書籍の貸し借りサービスの開発

IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3)

評価指標

各モジュールが正常に機能することを確認する

本研究の成果物を利用してもらい
5段階評価とコメントによるフィードバックを頂くことで指標とする