

学生選書支援オンラインシステムの構築 について

影林泰佳

目 次

1	はじめに	1
1	人の役に立つ Web アプリケーションを作る	1
2	本アプリケーションの利用対象者について	2
3	本アプリケーションの利便性について	2
4	同様のシステムの有無について	3
2	システムの構想について	5
1	システム構想	5
2	技術的な制限	6
3	作業計画	7
3	製作過程	8
1	サイト構成	8
2	一般利用者	10
3	登録利用者	11
4	リクエスト管理者	19
5	リクエストの検索機能	21
4	評価	22
1	アンケート結果	22
2	改良点と課題	23
3	自己評価	24
5	おわりに	25

1 はじめに

(1) 人の役に立つ Web アプリケーションを作る

大谷大学図書館⁽¹⁾には、学生選書プロジェクト（以下、sspj）と呼ばれる企画がおこなわれている。このプロジェクトは、「大学図書館には読みたい本が少ない」という学生の声により、「それでは学生の視点から図書館に置く本を選ぼう」という、当大学図書館の新しい試みである。

このsspjにおいて、学生が大学図書館に置きたいという意志を示す方法は2種類ある。1つ目は、学生選書コーナーに常備されている「図書リクエスト用紙」に記入し、併設されているリクエストボックスに投入する方法と、二つ目は、年に数回開催される「選書ツアー」と呼ばれる企画に参加する方法である。

私は選書ツアーには参加したことがある。しかし、リクエスト用紙によるリクエストをしたことがない。なぜなら、このリクエスト用紙を記入するには時間がかかるからである。

リクエスト用紙1枚につき、学生番号（利用証）・所属学部学科・氏名・希望理由・希望図書の書誌情報（書名・著者・発行所・発行年・価格・ISBN/ISSN）を記入しなければならない。この紙媒体からのリクエストは、リクエストの度に学生情報を記入し、書誌情報はインターネットなどを利用して取得しなければならない、という必要性がある。また、リクエスト後の検討結果は学生選書コーナーにて掲示されるため、自分自身で確認に赴かなければならないのである。

これらの3つの理由から、私はこのsspjでのリクエストは利用したいが手続きに手間がかかると感じたために利用したことがないのである。しかし、この点が改善できれば、もっと効率よく図書のリクエストすることができるのではないだろうか。具体的なシステム化について、

下記のようにまとめてみた。

- ・ 学生番号や氏名などを登録し、リクエストごとに記入する手間を省く。
- ・ 利用情報（リクエストの申し込み情報など）の確認を可能にする。
- ・ 書誌情報を容易に取得する。
- ・ 検討結果を直接本人に知らせる。

学生選書を利用してもらうことを目的とし、ひいては大学図書館をより多くの学生に利用してもらう契機として、sspjにおける図書リクエストのためのオンラインシステムを開発する。

(2) 本アプリケーションの利用対象者について

当大学独自の sspj に関するアプリケーションであるため、利用対象者は図書リクエストのサービスを利用する権利を有する者に限られる。ゆえに、当大学に在籍する学生のみにおいて利用されることを目的としている。

また、リクエストを集計する機能も設定したため、sspj を担当されている図書館職員の方と学生選書スタッフ関係者においても活用できるシステムになっている。

(3) 本アプリケーションの利便性について

オンライン上で図書のリクエストをうけるということは、書誌情報をデータとして利用できるということである。リクエストとして送られてきた書誌データの利用価値を次のようにまとめてみた。

- ・ 受け付けたリクエストをオンライン上で公開することができ、多くの学生に利用する契機を提供することができる。

- ・ 受け付けたリクエストの編集・削除など、リクエストデータの管理が容易である。
- ・ 受け付けたリクエストを対象に検索をすることができる。
- ・ 検索結果を指定する条件によって、絞込みや並び替えをすることが可能である。

1つ書誌データを作成することで、リクエストした本人・リクエストを集計や管理、審議を行う選書スタッフ・リクエストした本人以外の学生に活用することが可能である。

また、リクエストを行う際、希望する図書のデータを調べると同時にリクエストを送ることができるので、「利用者による図書のリクエスト用紙へ転記」→「担当スタッフによるデータ入力」の工程を省略することができる。リクエストしたデータも残るので、自分がリクエストを足した図書も忘れて、検討結果を見逃すことがない。さらに、自分がリクエストした図書の情報だけでなく、ほかの学生がリクエストした図書の情報も閲覧することができる。

学生選書スタッフのリクエストの管理においても、リクエスト履歴の確認・リクエストデータの検索・抜き出しや並び替えの応用など、利用者同様にデータを活用できると考えられる。

(4) 同様のシステムの有無について

考えついた2つのシステムまたはサービスから比較・検討した。

1つ目は現在、大谷大学図書館で利用されている「Web サービス⁽²⁾」である。これは利用登録を行えば、図書館サービスの一部をweb上から利用できる。具体的な内容は次のとおりである。

- ・ 個人貸出・予約・申し込み情報の確認と貸出中図書の予約。

- ・ 貸出情報と返却日を、登録したメールアドレスへ自動的に送信する。
- ・ (学外への文献複写の申し込み)

作成したアプリケーションに活用したい機能は、個人の貸出情報の参照と申し込み情報の閲覧、および貸出情報などを登録したメールアドレスに送信するという点である。(ただし、文献複写申し込みは利用する機会がなく、どのようなサービス内容であるかは比較できていない。)

従来用の紙によるリクエストでは、図書リクエスト用紙をそのまま所定のボックスに投入するため、リクエストした側に控えがのこらない。これでは積極的に s s p j に参加するほど、自分が希望した図書を把握しづらくなるのではと考えた。重複して希望を出す可能性もある。

しかし、この問題点は、上記の2点を応用することで解決することができる。送信したリクエストは、本人の意思により、確認とリクエストの状況を把握することが可能になる。また、利用者だけでなく、リクエストを管理する側においても、利用書からリクエストの状況への応答や重複への返答などの作業を省くことができるだろう。

現在のリクエスト方式では、リクエストする学生側と大学図書館側でやり取りがない。リクエスト者から大学図書館へ(リクエスト申し込み時)、大学図書館から利用者へ(リクエスト検討結果通知時)のお互いに一方通行な関係である。

だが、連絡先を登録することで、リクエストの審議が出たときのみならず、リクエストの内容に不備があったときになど、直接連絡を取り合うことが可能になるだろう。

2つ目は、当大学の卒業生が作成した「人文情報学科生のための書誌検索システム⁽³⁾」である。

このシステムは、オンラインショッピングサイト「Amazon.com⁽⁴⁾」

にて提供されている「Amazon Web サービス」を利用して作成されている。「Amazon Web サービス」とは、Amazon.com が所有している商品情報のデータベースを提供し、個人の要求に応じたアプリケーションの作成を支援するサービスである。作成されたシステムで可能なことを次にまとめる。

- ・ Amazon の商品情報を検索し、取得することができる。
- ・ 取得したデータは、製作者の意図どおりに配置することができる。
- ・ 人文情報学科生向けに授業別やよく利用するキーワードのカテゴリから検索することができる。

以上2つのシステムおよびサービスにおいて、利用もしくは応用できる機能があるが、活用目的が異なっているため、同様のシステムとは言い難い。2つ目のシステムにおいては、書誌情報を取得するという点で共通する目的ではあるが、本システムは、取得したデータをもとに、新たなサービスを提供することを目的とするものである。

2 システムの構想について

(1) システム構想

本アプリケーションは、下記の項目を有するシステム開発を目指したものである。

- ・ 利用者が希望するリクエストの書誌情報を取得し、そのデータをリクエストとして送信する。
- ・ 利用者から提供された書誌情報をもとに、リクエストを管理する。
- ・ 書誌情報やリクエスト情報は、リクエストを管理する側だけでなく、リクエストした本人・本人以外の利用登録者においてもオンライン上で公開し、利用できる。

- ・ 検討した結果などは、公開および E メールにて報告される。
- ・ 選書スタッフ内・登録利用者それぞれの間で意見の交換や交流ができる機能を設定する。

(2) 技術的な制限

「人文情報学科生のための書誌検索システム」で利用されていた「Amazon Web サービス」は、前述のシステム開発当時、「Amazon.com の商品データをサービスとして提供し、個人で作成するアプリケーションを支援する」ものであった。しかし、Amazon.com において提供方針に変更があった。名称も「Amazon アソシエイト（アフィリエイト）」⁽⁵⁾に変更され、アフィリエイトを利用する方を対象に商品データが提供されるということである。

アフィリエイトとは、ホームページやブログ・アプリケーションなどにおいて、データを提供するサイトへリンク貼り、そのサイトから商品が購入されると紹介料が得られるという仕組みである。

データは変わらず提供されているようだが、このアフィリエイト制度は本アプリケーションに取り入れるべきシステムではない。なぜなら、製作方針でもある人の役に立つアプリケーションの作成を目指しており、営利目的で提供するものではないからだ。

また、設定される提供先へのリンクは、商品ページへのリンクだけでなく、直接購入手続きへ進むリンクもある。書誌情報を取得し、そのデータをリクエスト管理者へ送信しなければならないが、このようなリンクがあれば、利用者に誤解や混乱を呼ぶ恐れもある。

「Amazon.com」の他に書誌データを提供しているサイトを探したが、全て有料などの条件が課せられており、書誌情報を取得する方法を

変更せざるを得なかった。

具体的には、別の画面で書誌情報を検索→必要な項目を入力してフォームで送信するという方法である。データを検索のみで取得するより手間はかかるが、検索結果と同時に表示されるリンクに戸惑うことなくリクエストすることができるだろう。

また、もう一つ構想から変更した点がある。リクエストの検討結果の報告等に活用する予定であった、利用者のメールアドレスの登録である。

直接結果の報告が届くので利用しやすいが、セキュリティ面に不安があったというのが理由である。このアプリケーションはオンラインで利用されるものであるため、様々な危険に冒されやすい。学内だけの利用であるが、ログイン時の認証も簡単な条件式のみであるため、万が一を考慮し、今回はこの設定を見送った次第である。

(3) 作業計画

1. 使用言語や機能について理解を深める
2. データベースを構成するテーブルの作成
3. システムトップページ
4. リクエストフォーム
5. 一般向けページ
6. 利用登録と個人ページ
7. 個人ページでの利用状況確認
8. 選書管理委員用ページ
9. リクエスト管理（編集・削除）
10. 検索・リスト表示・検討結果報告

11. その他、正規表現や追加機能

3年次で学んだプログラムの復習と、応用のために理解を深めた後、このシステムのメインであるリクエストフォームの作成からはじめる。データベースに保存しなければならないので、簡単にテーブルを作成していく。必要な項目ができれば、その都度更新するつもりである。リクエストの送信、登録、詳細ページまでを完成させる。利用者種別が、一般利用者（非登録利用者）・登録利用者・リクエスト管理者の三つに分けるため、それぞれページ分けとパスワードを設定する。登録利用者としリクエスト管理者のページは、対になる機能が多いため、機能の登録は極力平行しておこなう。

必要な機能がそろったあとは、利用登録・リクエスト登録などの場面において必要と思われる正規表現を追加したり、登録者がお互いにコミュニケーションを取れる機能またはページを作成する。

リクエストや利用登録、またログイン時など様々なパターンでテストを繰り返す必要がある。また、使用するページの量が予想されるので、エラーの発生や見落としに注意する。

3 製作過程

(1) サイト構成

このリクエストシステムは、メールのように希望する図書の情報を送るものではない。データベースを利用し、同じくオンライン上でリクエストを管理することができる。必要なデータを取得し、学生選書スタッフ側の管理ページで、受け付けたリクエストを確認したり編集したりすることができるのである。

このアプリケーションで構成してるデータベースのテーブルは以下の

とおりである。

- ・ request…リクエストの書誌データを保存するテーブル。
- ・ member…登録利用者の管理情報。
- ・ kiji…リクエスト管理者が発行するお知らせ情報の内容を保存する（お知らせ記事機能）
- ・ comment…リクエスト管理者画面において、リクエストの検討の際に、特定のリクエストについて意見を投稿することができる。（コメント機能）
- ・ web_clap…利用登録者向けのコメント。一言メッセージをつけることができる。（リクエスト拍手機能）

また、すでに記述があるとおり、このシステムでは三つの管理ページで構成されている。利用登録が済んでいても、ログインしない限りは一般利用者と同じ利用権限が与えられない。ログイン情報は cookie を利用しており、cookie を解除するには「ログアウト」するか、ウインドウを閉じれば解除される。

詳しいこのシステムでの cookie の導入については、登録利用者についての記述とともに行う。

- ・ 一般利用者（非登録利用者）：個人ページの登録申請を行っていない利用者向けページ。
- ・ 登録利用者：一般利用者画面ですでに利用登録が済んでいる利用者向け。ログインにより、個人ページに移動する。
- ・ リクエスト管理者：選書スタッフ向けページで、登録は不要である。管理者用ページのパスワードを入力することで管理ページに移動する。

(2) 一般利用者

一般利用者とは、この学生選書支援オンラインシステムへの利用登録の済んでいない利用者のことを総称する。この一般利用者は、選書スタッフからのお知らせ情報や、リクエスト検索などを確認できるのみである。最初はすべての利用者がこれに該当し、リクエストを申し込むにあたっては、必ず利用登録が必要である。

利用登録に必要な項目は、次のとおりである。

- ・ 個人ページにログインする際の学籍番号 (ID) ... 利用対象が当大学の在学生であるため、日常的に使用している学籍番号を採用した。
- ・ 名前 (ニックネーム) ... リクエスト投稿者の判断と、拍手機能を利用する際に個人を特定する時などに利用する。
- ・ パスワード... ログインの際の認証に使用する。半角英数 4 桁から 8 桁まで登録が可能である。

当システムのプログラム内では、利用登録の行われていない利用者、もしくはまだログインしていない状況の利用者はすべて「guest」という識別を行っている。例えば、リクエスト検索をかけた際に、利用登録者なら、拍手機能を利用できるよう、投稿フォームが表示されるが、利用者を識別している cookie 機能のデフォルト値である「guest」になっていると、投稿フォームが表示されないようになっている。

【例】 書誌情報詳細ページ (request_detail.php)

```
if($user!="guest")
```

```
//ここに上記の条件式に当てはまった場合の処理
```

[解説] \$user は、利用者の識別をしている変数。cookie を使用。サイトの index でデフォルト値として「guest」と保存するように設定している。管理者ページからログアウトした際は index へ戻るため、この

時点で「guest」の種別に変更される。

この条件式では、もし\$userの変数が「guest」以外なら以下の処理を実行するというものである。

(3) 登録利用者

(i) 登録利用者のログインについて

indexの「利用者登録」から登録が完了すると、すぐに左側のメニューの下方にあるID（学生番号）とパスワードを入力して利用できる。この入力フォームから送られたIDとパスワードが一致すれば登録利用者の個人ページを利用できる仕組みである。使用したsql文は下記のものである。

```
select user_id from member where member_id=$id and user_pass=$pass;
```

[解説] member テーブルの member_id 項目には、利用者のためのID（学籍番号）が保存されており、\$id という変数にログインページで入力された値が form より送られてくる。同様に、user_pass 項目には、登録されたパスワードが保存されており、\$pass 変数には入力されたパスワードが送られる。つまり、登録されたID（学籍番号）と入力されたID、登録されたパスワードと入力されたパスワードがそれぞれ一致し、さらに、そのIDとパスワードが登録されているユーザーのレコードを特定する user_id を取得する sql 文である。

この時に一致するレコードが得られなかった場合は、IDかパスワードが違っているか、まだ利用登録がすんでいないということになるため、もう一度入力を促すメッセージが表示される。

また、登録者の利用ページには、id を元に利用情報を呼び出すことが多々あるため、利用者の種類を識別する変数\$user に user_id (member

テーブルのレコード id) を保存したい。そのため、上記の sql 文にて ID (学籍番号) とパスワードが一致したユーザーの user_id を cookie 機能を利用した \$user という変数に保存し、このシステム内で利用者の種類を一度のログイン入力だけで判断することを可能にした。

この利用者を識別する \$user という変数は当システム全体で cookie 機能を利用している。ログインしていない、もしくは利用登録が済んでいない一般利用者を「guest」と設定するよう、index でデフォルトしている。登録利用者なら、member テーブルの user_id、リクエスト管理者なら、「sspj」がそれぞれ cookie に保存するように設定している。

これにより、例えば、利用者を識別する \$user が「guest」、つまり一般利用者だった場合に権限のない部分は表示しない、またはサービスが利用できないよう判断することができるのである。また、\$user が「guest」でも「sspj」でもない場合は、登録利用者の user_id が保存されているので、登録利用者の個人ページ以下において、\$user 変数に保存された user_id をもとに、リクエストした情報など、登録利用者個人の情報を取得するのにも利用できるのである。

(ii) 図書のリクエスト機能について

登録利用が完了すると、大学図書館に受け入れを希望する図書をリクエストすることができる。登録利用者メニューには、リクエストまでの流れを説明するページを用意している。その内容をまとめると次のようになる。

1. リクエストしたい図書は当大学に蔵書のないものであるかをチェックする。
2. 「amazon より本を探す」をクリックし、リクエストに必要な情報を取得する。(ただし、リクエストフォームの必須項目「書名」「著

者」「ISBN」がわかるメモや参考になるものが手元があれば、そのままリクエストフォームへ入力も可。）

3. リクエストフォームの項目を入力する。（タイトルや ISBN など間違いやすい項目は、コピーアンドペーストを推奨する。）
4. 確認ボタンを押し、次のページで入力不備などによるエラーやリクエストの重複によるメッセージがでなければ、送信ボタンを押して完了である。

基本的に、図書のリクエストは当大学図書館に蔵書がないものを希望するものであるので、利用者ページの左側のメニュー中ほどの「所蔵の有無を調べる」より、大学図書館の蔵書検索ページより確認するように促している。

リクエストする上での入力項目として、「書名」「叢書名（形式やシリーズなど）」「著者」「出版社」「発行年」「価格」「ISBN」「備考・希望理由など」の8項目を用意した。入力項目は以上だが、リクエストの送信と同時に、日付と利用者 id (user_id) が一緒に送信され、リクエストデータの項目として自動的に登録される。

このうち、「書名」「著者」「ISBN」を入力必須項目とした。この3つの項目があれば、ほぼ間違いなく図書が特定できるためである。よって、利用者とリクエスト管理者の間で相違が発生しにくい。また、当大学図書館に蔵書はないが、すでにリクエストされているものであるかを判断するための機能をリクエスト内容を確認するページにて設定している。そのため、3つの項目に当てはまるリクエストを発見した場合には、すでにリクエストがある旨のメッセージを表示することができ、リクエストの重複をさけることが期待できる。

入力不備によるエラーは主に2つのパターンがある。前述した必須項目に空欄がある場合と、必須項目のうちの「ISBN」コードに不備がある場合である。

ISBN コードのない本（流通に乗らないものや、ごく一部でしか利用されない図書、または郷土資料・パンフレット・自費出版物などの資料）は、入手しにくい可能性が高いか、もしくは非常に特殊なリクエストと判断される。こういった図書は特定するのが難しいため、必要な場合は、大学図書館で利用者側、大学図書館側双方でどのような本であるかを直接確認しながら特定していくほうが望ましいと思われる。本システムでは、正式な流通にのり、市場で購入の可能な一般に手に入る図書よりリクエストを受け付けることを前提としており、ISBN コードをリクエストするうえでの必須項目とすることで、リクエストをする側、受け付ける側の両者の間で混乱が生じないように考慮したものである。

また、この ISBN コードに不備がないよう、正規表現も設定している。ISBN コードは旧規格では10桁、2007年より現行されている規格では13桁で構成される。旧規格の ISBN コードは国際規格上では無効となっているが、書籍検索や注文などの場面においては、10桁の ISBN も併用できるよう便宜が図られている場合がほとんどである。

本システムで書誌情報の参照先としてリンクを指定している「Amazon.com」においても、10桁の ISBN コードの割り当てられたものは、10桁、13桁ともに併記されている。こちらの書誌情報を参照し、リクエストする場合は13桁の ISBN コードを入力してもらうよう推奨しているが、個人で用意された情報源においても対応できるよう、ISBN コードの正規表現の中では10桁または13桁の0から9までの半角数字であることを指定している。また、ISBN コード内に含まれている「-

(ハイフン)」を取り除く正規表現も加え、統一した形式で表示することが可能にした。

ISBN が間違いなく入力されると、次は重複するリクエストがないか確認するように。条件は、(1)「ISBN が一致するもの」(2)「タイトルと著者が一致するもの」である。

この条件を設定するには、まず必須項目だけで処理しなければならない。利用者の最低限の情報でも対応できるように設定している必須項目は、空欄のものがなく、この情報だけでも希望する図書が特定できるため、逆にいうと、この3つの項目だけで重複があった場合も特定できるということである。

条件の1つめは、特定の1冊しか示すことのない ISBN コードであるので、まったく違う ISBN コードを入力された場合を除けば確実に重複の発生が確認できる。2つめは、出版の形式が違った場合を考慮しての条件である。例えば、単行本と文庫本である場合などだ。出版の形式が違えば ISBN コードは異なるので、ISBN コードが一致しなくても、タイトルと著者が同じであれば、それは同じ本である可能性が非常に高いと確認できるためである。

これら2つの条件のいずれかに当てはまるリクエストがあれば、リスト形式で抽出し、利用者に向けて同じ図書であるか確認を促すメッセージが表示される。抽出されたリストの項目は、「タイトル」「著者」「ISBN」の3つであるが、タイトルに書誌詳細ページを開けるようリンクを指定している。もし同じ本であった場合は、リクエストの送信内容を確認する画面で取り消すことができるので、まったく同じ本のリクエストが重複して送信されることが少なくなるだろう。また、リクエストのデータが抽出されたとしても、リクエストを送信することが可能に

なっている。例えば条件2によって抽出されたもので、単行本がすでにリクエストされていたとする。しかし、「内容が追加されて文庫で出版されているから」などの理由から希望するのであれば、そのようにリクエストしても問題ないのである。図書館の本は耐久性の面で、文庫本より単行本の購入を推奨しているため、このような例も起こりえると考えられるだろう。

(iii) リクエスト状況の確認機能について

リクエストしたデータのリクエストは適宜受け付け状況確認することができる。これは登録利用者メニューにある「リクエスト状況の確認」にあたる機能である。リクエストされた図書が受け入れられるまでの段階下記のように設定した。

- ・ 受付中... リクエスト管理者によって、まだリクエストの図書が確認できていない状況を示す。
- ・ 検討中... リクエスト管理者によって、リクエストの図書が確認され、学生選書スタッフにおいて検討されていることを示す。
- ・ 購入予定... 学生選書スタッフの検討により、購入が決定されたことを示す。
- ・ 保留... 学生選書スタッフの検討により、購入が保留とされたことを示す。
- ・ 購入済み... すでに学生選書として購入されたものであることを示す。
- ・ その他... 上記以外の特殊な状況であることを示す。

この「リクエスト状況の確認」画面では、よく利用する場合も考慮し、状況別にリスト化している。さらに、日付からリクエストを絞り込むことも可能である。

また、送信したリクエストを取りやめたいときは、リクエスト状況が「受付中」の場合に、自らキャンセルをかけることが可能である。しかし、すでに次に記す「リクエスト投票」がかけられていた場合は、そのリクエストに支持者がいるとみなし、リクエスト者本人であっても、キャンセルのコマンドが表示されないように設定している。

リクエスト状況は書誌詳細ページにも表示されている項目であり、全ての閲覧者において確認できる。よって、自分のリクエストの状況だけでなく、書誌検索などから書誌詳細ページを閲覧した場合にも、その図書の状況を確認することが可能である。

(iv) リクエストへの投票機能

リクエストの重複を確認する機能を取り入れたのは、同じ内容のリクエストが発生した場合に、データを共有することができないからである。ゆえに、別の利用者が同じ図書に対してリクエストをすれば、同じデータがあるたびに、繰り返し内容を確認する必要がある。リクエスト管理者側の負担を考慮し、リクエストの重複を避けるようもうけられた機能である。

しかし、リクエスト状況において確認できるのは自分のリクエストした図書のみである。よって同じリクエストがあれば、同じ本を希望する利用者はリクエスト状況を、個人ページで確認することができない。そこで加えた機能は「リクエストへの投票機能」である。すでにあるリクエストの書誌データに対して「投票」を行うと、個人ページで希望するリクエストの受付状況を確認することができる。また、希望理由も意思表示できるよう、一言程度のコメントを投稿できるように設定している。

この機能は、書誌詳細ページにおいて、登録利用者だけに与えられた

サービスである。書誌詳細ページはどのリンクからも共通して利用されているページであり、このページ内に利用者種別を表す\$user の値が「guest」または「sspj」以外の利用者の場合のみに表示されるように設定されている。

例えば、リクエスト確認画面において、重複条件で抽出された一覧には、書誌データを確認するリンクがある。同じ本のリクエストであった場合でも、リンク先の書誌詳細ページにてコメントをそえて投票すれば、他の利用者が送信したリクエストの状況も確認できるのである。

投票が可能なリクエスト状況は、受付中から購入予定までである。当初は、「検討中前までのリクエストキャンセル」において、リクエスト者本人がリクエストを取りやめた場合、リクエストの支持表明であるコメントも消えてしまうことを考慮し、「検討中以降のリクエスト」を対象にした機能にするつもりであった。しかし、それではリクエストの重複が抽出された場合に、まだそのリクエストが「受付中」の状況だと、この機能が利用できない可能性があることが判明し、前述のとおり、「投票によって支持されているリクエストはキャンセルができない」という方針にいたったのである。

アンケート形式や、いわゆる「お気に入り機能」的な方式のほうが、効率がよいかとも考えた。だが、一言でも反応があるほうが、読みたい本を探すときや、他の閲覧者にとってもただのシステムではないという印象をあたえられる。同じ大学の学生同士、このようなコミュニケーションも出会いの役に立つアプリケーションになればと考え、この機能を取り入れたのである。

(4) リクエスト管理者

(i) ログインについて

リクエスト管理者、つまり当大学図書館司書および学生選書スタッフを対象としたページである。ログインは、本システムのタイトルの下にあり、パスワードの入力だけで利用できる。自ら学生選書スタッフとして参加したことにより、それほど大人数の不特定多数を対象とするわけではなく、作業や閲覧内容が共通する項目が多いことから、ID とパスワードを発行する個人ページ的な方式より効率がよいと考えたためである。

(ii) リクエスト管理

利用登録者より送信されたリクエストは全てメニューから「リクエスト管理」のページで閲覧できる。表示は利用登録者のリクエスト状況確認画面ページのようにリクエストの状況別に分かれている。

まず、リクエスト管理者はリクエスト内容の確認をしなければならない。「受付中」一覧には、リクエストの日付が昇順にソートされている。表示されている項目は、「受付日」「書名」「著者名」「リクエスト者 ID」「編集」「削除」である。「編集」をクリックすれば新しいウィンドウが表示され、左側にリクエストの編集画面、右ページには書誌情報を参照するための「Amazon.com」のサイトへ接続される。ここで入力されていた ISBN コードを利用して検索をかけ、入力内容の確認と未入力項目を補い、受付状況を「検討中」に変更する。

最初の「受付中」項目を確認・編集する方法は記したとおりだが、後に入力ミスを発見したときなども、この編集画面から同様に行う。また、この編集画面は、「リクエスト管理」の「受付中」一覧だけでなく、全ての状況別一覧にも表示されている。さらに書名を選択し、書誌詳細ページに移動した先でも、\$user がリクエスト管理者を示す「sspj」で

あるため、編集と削除を行うためのリンクが表示される。もちろん編集や削除権限のない一般利用者と利用登録者には表示されないように設定されている。

リクエスト状況が「検討中」に移行したリクエストは、「検討中リクエスト一覧」という学生スタッフ向けのページにも表示される。ここでは、「検討中」リクエストをスタッフが確認する目的がある。ほかのページ同様、リスト形式で表示され、書名を選択することで書誌詳細ページへ移行する。登録利用者には投票の機能があるが、リクエスト管理者には、その書誌について意見が述べられるよう、通常のコメント機能が設置されている。これもユーザー権限において、表示・非表示されるものである。

(iii) 選書スタッフ内での検討と検討結果

受け付けたリクエストの確認が終わると、学生選書スタッフを交えて受け付けたリクエストの検討に移る。

選書スタッフの活用方法案をまとめてみた。

1. 受付中の資料は、一定期間を決め、該当する日付の分のみデータを確認する。
2. 検討中に移ったリクエストを学生選書スタッフに目を通してもらい、コメント機能を使って意見交換を行う。
3. 検討が終わったリクエストは「購入予定」か「保留」に受付状況を変更する。

コメントの機能は、「検討中一覧」の書名から書誌詳細情報へリンクを設定しているので、そこで利用してもらおう。ここでも利用者権限により、選書スタッフの場合はコメントの入力フォームが表示されるようになっている。

このように、オンラインで意見交換を行うことができるため、効率よく検討を行う場を提供することが可能である。

リクエストの検討が終わると、この時点リクエストした側へ検討結果の報告を行うべきである。しかし、利用者のメールアドレスの登録を取りやめたため、リクエスト側には、受付状況が変わったことで、検討結果を示すにとどまっている。

(iv) お知らせ・コメント管理など

その他にも、更新履歴や、新しくリクエストの本が購入した時の連絡をシステムの index に表示できるようにするために、「記事管理」機能がある。学生選書スタッフ以外の利用者を対象にしたお知らせを表示するために利用するものである。記事の削除に関しては、4桁の数字からなる「削除キー」を設定し、記事管理画面の記事一覧において、削除と変更はこのキーを利用する。これにより、複数いるリクエストの管理者であっても、作成した記事に無断で改ざんや変更をできないようにしている。

また、記事の管理と同時に、投票コメントの管理もこの管理者に権限があり、一覧を取得することや、不適切な投票コメントなどが発見された場合は、こちら側から削除することができるようになっている。

(5) リクエストの検索機能

全システムの利用者に共通して利用できるサービスは、「リクエストの検索」機能である。

この検索の対象は、リクエスト状況が「受付中」のものからすでに購入済みとなった図書などを含める。つまり、このシステムにリクエストされた全てのデータを対象に、リクエストの検索をすることができる。

また、特定の検索語句がなくても、リクエストの受付日を日付から取得することができたり、リクエスト状況から検索することもできる。

このように、ただ特定のデータを検索するだけでなく、漠然とデータを眺めることができるので、読みたい本を探すときや、読みたい本が見つかったときに、すぐに投票機能を利用することができる。

4 評価

(1) アンケート結果

作成したアプリケーションのベータテストを、三回生のゼミ生9人をお願いした。結果は次の通りである。

「見やすさ」... ほぼ全員から高評価で、シンプルで見やすいとの意見をいただいた。

「使いやすさ」... こちらもそれほど悪くはなかったが、(1) パスワードを直接表示されていること (2) リクエストフォームの諸注意がわかりにくい (3) 利用者の解説ページが見あたらない、などの意見をいただいた。反省点は、機能を優先して作成していたため、せっかくのベータテストの機会での利用説明がきちんと作成できていなかったことである。

2つ目のリクエストフォームの諸注意については、注意を促す印はあるものの、解説がなかったこと。利用説明にもあればいいが、ISBNコードの13桁を推奨する旨もフォームの項目に併記した方がわかりやすいとの意見である。見慣れたシステムだと気づかなくなってしまうがちな細かなところも指摘をいただき、他の点でも「初めて利用する利用者」の立場になり、もう一度全体を確認することができた。

パスワードに関しては、HTMLのフォームの部品に「パスワード」と

呼ばれる部品が存在していたことがわかった。この部品を取り入れることにより、問題は解決した。

「もっと他に機能があったらいいと思うか」... 具体的な意見はなかったが、「そう思う」と答えた意見が多かった。この意見をとらえ、このシステムというものにコミュニケーションを持たせようとこの後に付与したのが、お知らせの「記事機能」とコメント投票である。

(2) 改良点と課題

「人の役に立つ Web アプリケーションを作る」という方針でここまで作成したが、改めて制作者の目線ではなく、利用者の立場から利用してみると、至らない点がある。

まず第1に、リクエスト検索についてだが、これは読み仮名検索ができない。読み仮名検索を可能にするには、書誌情報を登録する request テーブルにおいて、読み仮名の項目を用意しなければならない。また、書誌データを取得するものではないので、リクエスト管理者によって入力する必要がある。大勢に利用されることになれば、リクエスト管理者側に負担がかかると考え、断念した。

しかし、著者名から検索をする際など、一カ所でも間違いがあれば結果が出ない。そのため、著者名の読み仮名はわかるが漢字がわからない時などにも検索できる必要があるだろう。

また、もっと詳しく検索できるように複合検索やキーワード検索・カテゴリ検索などもあれば、効率よく書誌情報を取得したり、利用者が支持するリクエストの発見につながるのではないだろうか。

第2に、当初構想していた、Eメールを利用した検討結果の報告である。

個人が特定されるという危うさと、グローバルな E メールを利用することで、学外への誤送信も起こりうる。学内用のアドレスを利用できればよいが、正規表現の必要もある。アドレスの変更や送信エラーの際の対応など、この機能を追加するには、さらに時間を費やす必要があるだろう。

(3) 自己評価

作成当初から、重要な点の方針転換があった。一番最初に考えていたものと比べれば、機能性が大きく異なる。しかし、一つ一つの機能については満足できたものがいくつかある。「書誌詳細ページ」は特に力を注いだうちの一つである。

書誌詳細ページはこのシステムにある 3つの権限が全て混ざり合う複雑なプログラムになっているのである。組み方は少し雑かもしれないが、私がしたいと思うことを全て実現できた 1 ページである。この他にも、権限が混ざり合うページや機能は特に時間をついやし、おおよそ満足できている。

リクエストを受ける、受けたデータを活用する、リクエストの状況を管理する。これらのことは、学生選書プロジェクトの図書リクエストをオンラインでできるようにしよう、と考えてから、最低でも実現させたかった機能なのである。

しかし、作成を続けているうちにこのことを忘れてしまっていたように思う。なぜなら、「こんな機能をつけたい」「こうすればもっと上手くいく」など、プログラムをくんでいくにつれ、できることならやると、様々なことに手を出していたからである。

改めて見直してみると、力を入れたところは満足できている。だが、

制作当初に考えていなかった機能がたくさんある。あった方が便利だと思う機能ではあるが、中途半端に取り入れたせいで、特に正規表現や細かな機能など、まだ満足できていない部分も残っている。

もっとシンプルで、「こんなことができるのだ」と堂々といえるシステムをつくれたらもっとよかったと考えた次第である。

5 おわりに

この研究への取り組みは、ただプログラムを使ってアプリケーションをつくることではなかった。結果的に、私はこのことを勘違いしたまま、効率的な「便利さ」だけを求めていたように思うのである。

参考にさせていただいた先輩「人文情報学科生のためのオンライン検索システム」は、まさに私たちにとって「これは便利だ」というところを的確に捉えて実現されたものであろう。

人の役に立つアプリケーションを作るは、万人に必要でなくてもよいのである。本当に必要な人にとって役に立てばよいのである。

私が出られたのはこのようなことでしかないが、次に機会があれば、もっとよりよく人の役にたてるアプリケーションをつくれるという自信を持つきっかけになったと思うのである。

注

- (1) http://www.otani.ac.jp/kyo_kikan/library/index.html
- (2) http://www.otani.ac.jp/kyo_kikan/library/nab3mq00000016zu.html
- (3) 「人文情報学科生のための書誌検索システムの作成-Amazon Web サービスをりょうして-」(2006 年度卒業論文、梶木美奈さん)
- (4) <http://www.amazon.co.jp/>
- (5) <https://affiliate.amazon.co.jp/>

文献表

アंक

2007 『PHP の絵本 Web アプリ作りが楽しくなる 9 つの扉』翔
泳社

谷尻かおり

2006 『これだけはおさえないデータベース基礎の基礎 設計と運用
の基本から SQL の書き方まで』技術評論社