

# 学生選書システムの実用化にむけて

太田悠介

## 目 次

1	序論	1
1	はじめに	1
2	学生選書システムの役割	2
3	同様のシステムの有無	3
4	システム引継ぎ後の改善点	3
2	企画構想	4
1	システム改善構想	4
2	サーバーへのアップロード準備	6
3	Product Advertising API の導入検討	6
4	拍手・投稿コメント機能の見直し	7
3	制作過程	8
1	作業工程	8
2	お知らせ関連の製作	9
3	リクエスト一覧の改善	10
4	拍手機能と投稿コメント	12
5	ソートの実装と問題点	16
6	一括編集・削除機能	18
4	評価	19
1	アンケート結果	19
2	改善点と今後の課題	22
5	結論	23

## 1 序論

### (1) はじめに

学生選書システムとは、大谷大学図書館で行われている学生選書プロジェクト(以下:sspj)という取り組みを、より多くの方々に利用してもらおうと昨年の卒業生によって「学生選書支援オンラインシステムの構築について<sup>(1)</sup>」という名目で開発されたシステムである。今回、この学生選書システムを実用可能な段階にすることで、図書館の利用者・sspjスタッフに少しでも利用しやすい環境を提供したいという思いから、この学生選書システムの開発を引き継ぐことにした。

このsspjとは、読みたいと思う本を学生自らがリクエストし、その本を図書館の特定のコーナーに置くことで、より多くの人に図書館を利用してもらおうという試みである。従来のsspjでの図書リクエスト方法は、専用のリクエスト用紙に必要事項を記入し、図書館に設置してあるリクエストボックスに投稿する方法と、選書委員という図書館が学生に対して独自に募集しているメンバーの一員になることで、「選書ツアー」と呼ばれる催しに参加して本を選ぶという方法の二種類が存在する。しかし、これらのリクエスト方法は、それぞれ問題点を抱えている。

まず専用のリクエスト用紙に記入する場合、学生自身が希望する図書の題名・著者名・出版社などの情報だけでなく、ISBN(国際標準図書番号)という図書を特定するための番号を書き込まなければならない。このように様々な情報の記入を求められるので、その場でリクエスト用紙を記入し投稿するということが難しく、一度用紙を持ち帰ってインターネットなどを利用し情報収集する必要がある。さらに、自身のリクエスト結果を確認するには、特設のコーナーまで掲示を確認しにしなければならない。

もう一方は選書委員のメンバーとなり選書ツアーに参加するという方法だが、まだまだ知名度が低く、私自身もこの卒業制作に携わったことで初めて選書委員という存在を知ったというのが現状である

これらの問題点を少しでも解消し、多くの学生が気軽に sspj の制度を利用できるように学生選書システムの開発を引き継いでいこうと考えた。

## (2) 学生選書システムの役割

学生選書システムでは、学生は利用登録を行うことでリクエストの申請・sspj スタッフからのお知らせ確認・申請済みリクエストへの投票機能などを web サイト上から利用することができる。

web からのアクセスを可能にしたことによって、図書館まで足を運び投票用紙に記入する手間を省くことができると同時に、Amazon<sup>(2)</sup>から書籍情報を入手しながらリクエストフォームへ入力できるため、事前に書籍の情報収集をする必要がなく入力ミスも防ぐこともできる。

さらに管理者である図書館スタッフ専用の選書委員会管理メニュー(以下: sspj メニュー)も作成したことにより、図書館スタッフの方々がリクエストの確認・リストアップ・発注などの作業を効率よく行うことができるようになった。

このように学生選書システムはリクエストの申請を web で記入・管理することにより、学生は自宅や学校のパソコンから気軽に希望する書籍のリクエストが可能になり、sspj の目的の一つでもある図書館の利用人数増加に繋げることができるシステムである。選書システムの簡単な使用方法については、資料編に「学生選書システム使用方法」として記載しておいたので参考にしてほしい。

### (3) 同様のシステムの有無

上記でも述べたように、この学生選書システムは昨年度の卒業生が製作したシステムを引き継ぐ形で製作してきた。今回新規にシステムを作成するのではなく、引継ぎという形を選択した理由として、二つの理由が挙げられる。

一つは、まったく新しい選書システムを構築するには膨大な時間が掛かってしまうということだ。今回の製作のテーマとして、選書システムをただ構築するだけではなく、実用化することを一番の目標として掲げている。実際に学生や図書館スタッフの方に使用してもらうことで、「人に役立つものを作る」というゼミのテーマに沿ったシステムを開発できると感じたため、新しくシステムを構築するのではなく、引き継ぐという形で実用化に向けて製作に取り組むことにした。

もう一つの理由は、この学生選書システムに利便性と実用性を見出すことができたからである。引き継いだ段階では、レイアウトや機能面において、不十分な箇所は多々見受けられたものの、web上でリクエスト申請するというアイデアや、実用化した際に学生や図書館スタッフ双方に役立つシステムになると感じたことが引き継ぐきっかけとなった。

### (4) システム引継ぎ後の改善点

昨年の卒業生から学生選書システムの引継ぎを行った時点では、図書館スタッフの方々はシステム自体を実際に見てはいないとのことだった。さらに作成途中の機能、サーバーへのアップロード等も行われていなかったため、これらの問題を含め次のようにシステムの改善点をまとめてみた。

- ・ システムをサーバーへアップロードし、web上で使用可能にする。

- ・ Product Advertising API (Amazon アソシエイト) 機能を導入する。
- ・ 全体的なレイアウトの見直し。
- ・ 申請済みリクエスト・投票コメントの管理方法見直し。
- ・ コメント機能の制限と簡略化する。
- ・ sspj メニューを改善する。
- ・ その他機能を追加する。

上記の項目を中心にシステムの改善を行い、図書館での実用化を第一に考えながら製作した。

## 2 企画構想

### (1) システム改善構想

選書システムを引き継いだ際、まず目に付いたのが膨大な数のファイル数である。言語は php がメインで使用されており、css によってデザインが形成されていた。過去に授業で学習した内容を復習しながら、まずはこれらの膨大なファイルが形成するページを一つひとつ確認し、システム構成を理解するところから始め、改善点を見出していくことにした。このシステムで使用されているデータベースのテーブルは、

- ・ request：申請されたリクエストに関するデータを保存する。
- ・ member：利用登録した学生の情報を保存する。
- ・ kiji：管理者が投稿したお知らせのデータを保存する。
- ・ comment：投稿コメントのデータを保存する。
- ・ web\_clap：拍手のデータを保存する。

以上5つとなっており、これらは変更せずそのまま引き継いで使用していくことにした。

システムを使用して見た全体的な感想として、機能は追加したが、その後使用された時の状況までは想定できていないといった部分が多数見受けられた。これらの改善点を実際にシステムを動かし、確認しながら製作していくことで、使用者の立場に立ったシステムに仕上げていった。

ページレイアウトはフレームによって製作されており、以下のフレームで構成されていた。

- ・ title(上部のページタイトル)
- ・ menu(左のメニュー)
- ・ mein(中央のメイン)
- ・ footer(下部のコピーライト)

これらの4つの基礎フレームレイアウトは変更を加えず、残したままシステムの改善に取り掛かった。(資料編：図1 [フレームレイアウト] 参照)

学生が新規会員登録することで利用できるマイページでは、ご利用案内・リクエスト方法などを閲覧しやすく変更し、メニューの項目も理解しやすい用語で記載することで、利用の仕方がわからないということを防ぐ。sspjメニューも管理者に負担をかけないように、学生から送信されたリクエストなどの情報を一括で編集・削除等ができる項目を設け、シンプルなデザインに変更した。

以上のように、主に見やすい配色・使いやすさ・細かな機能面を中心に、パソコンに慣れていない学生でも使いやすいよう改良を加えていくことを目標として改善していった。

## (2) サーバーへのアップロード準備

今回選書システムを web 上で使用可能にする事を目標としているため、学内のサーバーへアップロードしなければならない。まず大学の情報アシスタントの職員の方に協力を仰ぎ、アップロード作業を進めていく。その後、ファイルをアップロードするために必要な FTP ソフトである「Cyberduck」の登録を行い、すべての作業が完了した。アップロード後は、ファイルを更新し Cyberduck にアップロードするという手順を繰り返し行うことで、システムを徐々に改善していった。

選書システムにアクセスするためのアドレスは、

- ・ [www2.otani.ac.jp/fkdsemi/0748020/sensho/index.php](http://www2.otani.ac.jp/fkdsemi/0748020/sensho/index.php)

となり、上記のアドレスをブラウザに記入することで、インターネットに接続できる環境ならば選書システムを使用することが可能となった。

## (3) Product Advertising API の導入検討

選書システムを引き継ぐにあたって、Amazon が提供する「Product Advertising API (Amazon アソシエイト) サービス<sup>(3)</sup>」の導入を検討してみた。

Product Advertising API サービスとは、専用のアカウントを取得することで、Amazon から様々な書籍データを取得することができるサービスである。

当初の予定では、Product Advertising API サービスを利用し、選書システムを通じて Amazon の商品データや販売情報の取得することで、現在のリクエスト申請フォームを改良することを目標としてきた。しかしサービスの概要を読み、アカウントの登録などを行っていく中で、Product Advertising API サービスを利用するには「Amazon アソシ

「アソシエイト・プログラム」への登録が必要ということがわかった。

Amazon アソシエイト・プログラムとは、web サイトなどに Amazon の商品データを表示する広告を貼り、ユーザーがそれをクリックして表品を購入した場合に、web サイトの管理者が Amazon から紹介料を獲得できるというしくみである。このサービスを利用するためには、Product Advertising API サービスとは別にアカウントを取得する必要があると同時に、クレジットカードの登録も必要とされていた。

この時点で、利用するためのハードルの高さ、さらに大学図書館で利用するシステムに金銭が絡むサービスを導入することは似つかわしくないと感じ、今回は導入を見送ることに決めた。

#### **(4) 拍手・投稿コメント機能の見直し**

選書システムには、自分もしくは他人が申請したリクエストにコメントできる「投稿コメント」機能と、同じくリクエストに一言のコメントを付けて、リクエストをお勧めすることができる「拍手」機能が導入されている。

これらの機能はどちらも似たようなコメント投稿システムのため、利用者の混乱を招く可能性がある。現に私が選書システムを使用した際も、同じような機能が二つもあるためどちらを使用しているのか分からなくなってしまった。

これらの機能は、以前は不可能だった管理者と学生との意見交換を可能にする重要な機能のため、もう少し分かりやすい方法でコメント機能を活用できるよう工夫した。その結果は後述の「製作過程-拍手機能と投稿コメント」で詳しく説明する。

### 3 制作過程

#### (1) 作業工程

製作については TOP ページ、マイページ、sspj メニューという 3 つに分けて、以下の内容でシステムの改良を行っていく。

##### (i) TOP ページの改善点

1. 「ご利用案内」の見直しを行う。
2. 「新規利用登録」に E-mail 記入欄を追加する。
3. お知らせのテーブル化、過去のお知らせをリスト形式に変更する。
4. 新着リクエストを「最新 20 件のみ表示」から「すべて表示し 20 件以上は次へボタンを押して表示」形式に変更する。
5. リクエストの投票数のカウントを製作する。

##### (ii) マイページの改善点

1. 「登録内容変更」でニックネームのみの変更に対応させる。
2. 「新着リクエスト」に投票ボタンを追加する。
3. 投稿コメントフォームを廃止し拍手コメントのみに絞る。
4. 新着リクエストを「最新 20 件のみ表示」から「すべて表示し 20 件以上は次へボタンを押して表示」形式に変更する。
5. 「送信済みリクエストの状況確認」のリクエスト取り消し方法を変更する。
6. 「リクエスト一覧」にソート機能を追加する。

##### (iii) sspj メニューの改善点

1. 「お知らせ管理」を追加する。
2. 「登録者管理画面」を追加する。
3. 「新規お知らせ投稿」の本文入力欄に改行タグを追加する。
4. リスト表示するページ (リクエスト一覧、投票コメント管理) に「次

の20件に」ボタンを追加する。

5. 「リクエスト一覧」にチェック項目の一括編集機能を追加する。
6. 「投票コメント管理」にチェック項目の一括削除機能を追加する。
7. 「投票コメント管理」にソート機能を追加する。

上記の改善点を中心に作業しながら、全体的なレイアウトの変更も同時に行っていた。

## (2) お知らせ関連の製作

まずは管理者が学生に対して行う「お知らせ」メニューを製作していると思う。TOP・マイページでのお知らせはどちらも同じ使い方で閲覧でき、sspjメニューにはお知らせを投稿するための「新規お知らせ投稿」と、編集・削除を行う「お知らせ管理」を新たに設けた。(資料編：図12 [お知らせ管理] 参照)

初期のお知らせ確認ページはテーブルの幅が固定されておらず、記事の長さによって枠が小さくなったり大きくなったりしてしまっていたため、記事によってテーブル幅にばらつきが出て見づらくなっていた。

そこで table タグに

```
table width="サイズ" border="0" table-layout: fixed;"
```

という記述を追加して幅を固定したことにした。これにより、改善前より記事ごとのバラつきが無くなり、記事内容を分かりやすく表示できるようになった。(資料編：図3 [お知らせ] 参照)

次に過去に投稿した記事が、最新の記事と同じページに表示されるといった問題を改善する。以前は投稿したお知らせが全てひとつのページに表示されていたことから、最新の記事がどれなのか分かりにくい状況となっていた。

そこで、お知らせのトップページには最新のお知らせのみを表示させ、過去のお知らせに関しては別ページにリスト表示させる方法に変更した。お知らせページを最新記事と過去記事にファイルを分けたことで、閲覧する際も分かりやすくなったように思う(資料編：図4 [過去のお知らせ] 参照)

sspjメニューには、新たにお知らせを編集・削除することができる「お知らせ管理」を追加した。ここでは、新しいお知らせを投稿する際に入力する四桁の登録キーが必要となる(何も入力しなかった場合は空白でキー送信)。これらのキーを入力することで初めて編集・削除を行うことができるため、もし投稿者以外の人間が記事の管理を行おうとしても、できない仕組みとなっている。

### (3) リクエスト一覧の改善

次に申請されたリクエストを確認することができる「リクエスト一覧」(TOP ページ・マイページ・sspjメニュー)のレイアウト変更と、新しい機能を追加するための改善に取り掛かることにした。(参考ファイル：newbook.php)

初期のリクエスト一覧は、最新のリクエストを20件のみしか閲覧できないよう設定されていたため、20件を超えた場合利用者は過去のリクエストを見ることはできなかった。この問題を解決するためには、まず「次の20件」というボタンを作成し、指定の件数を超えた場合は次のページへ移動してリクエストを閲覧できる仕様に変更しなければならない。

まずはすべてのリクエスト件数をカウントする\$all\_countという変数を作る。その後、1ページ内に表示するリクエスト件数をカウントす

る変数を”\$count”と指定し、\$count に何も入っていない場合は\$count=0 となるように、基礎を作っていく。

```
【例】 if ($count == "") {  
    $count = 0;}  
}
```

上記の基準を元にページを先送りにするプログラムを記述していくのだが、その際 SQL に数件 (自分で好きな件数を指定) のみ検索して表示させるという以下のコマンドを付け加える事で、リクエスト一覧の表示件数をコントロールすることができる。

```
【例】 $sql = "$sql limit $count, 件数入力”;
```

あとはこれらの変数を if 文に当てはめ、1 ページ内のリクエスト件数が指定した件数に達した場合、次のページへ移動するというプログラムを記述していく。このプログラムによって、最新 20 件しか表示されなかったリクエストが、過去に送信されたものも閲覧することが可能となった。(資料編：図 5 [新着リクエスト一覧] 参照)

このページの先送り機能は、リクエスト一覧だけでなく、

- ・ 過去のお知らせ一覧。
- ・ 送信済みリクエストの状況確認 (マイページ)。
- ・ お知らせ管理 (sspj メニュー)。
- ・ 投票コメント管理 (sspj メニュー)。

以上のページでも導入し、閲覧しやすくすることができた。

リクエスト一覧には、以前からあった拍手機能を改良した投票コメント機能も実装したが、これについては後の「拍手機能と投稿コメント」で詳しく説明したいと思う。

#### (4) 拍手機能と投稿コメント

選書システムには、拍手と投稿コメントという二つの機能が実装されていた。これらの機能は、利用者である学生や、管理者である図書館スタッフが意見交換できる重要な機能のひとつである。それ故に改善点も多いので、力を入れて製作していきたいと思う。

##### (i) 機能の違いと共通点

初めてシステムを使用した際には、どちらの機能も同じように思えて違いが分からなかったのだが、プログラムが記述されている php ファイルを何度か確かめるうち、その機能の違いと使い道を理解することができた。

まず投稿コメント機能だが、これは投稿したリクエストに比較的長いコメントを付けることができ、利用者である学生、そして管理者である図書館スタッフの双方が利用できるコメント機能であることが分かった。投稿フォームには投稿者名と本文を記入することができ、送信されたコメントはリクエストのタイトルをクリックすることで表示される「書籍データ詳細」で確認することができる。しかし、コメントの表示数が指定されてないため、多くのコメントが送信された場合、果てしなく下までスクロールしなければいけない状態だった。

次に拍手機能だが、投稿コメント機能とは違い投稿者名を入力するフォームはなく、一言程度のコメントを送信ができる。sspj メニューに個別の「拍手管理」メニューが作成してあったことから、拍手はリクエストされている書籍に一票投じる投票機能であることが分かった。これも「書籍データ詳細」に送信されたコメントが表示されるようになっており、表示件数に制限は設けられていなかった。

これらの二種類の機能は多少の違いはあるものの、

- ・ 機能名がややこしく混乱を招く。
- ・ どちらもコメントを投票する機能である。
- ・ コメントの管理ページは拍手機能のみ。
- ・ 投稿されたコメントが閲覧できるページが同じ。

こういった点から、投稿コメント機能は削除し拍手機能のみを改良して実装することに決めた。

## (ii) 拍手機能の改善

まず拍手機能という機能名が誤解を招きやすいと感じ、「投票コメント機能」という名前に変更した。それに伴い、sspj メニューの「拍手管理」も「投票コメント管理」へ変更し、管理形式もリスト表示に変更した。

機能面の変更だが、以前はリクエストタイトルをクリックした「書籍データ詳細」内で投票・確認することができたが、投票フォームの場所が分かりづらいという問題を抱えていた。そこで、リクエスト一覧のテーブルに新たに投票欄を設け、様々なリクエストを実際に確認しながら投票コメント機能を使用できるよう改善し、同時に投票コメントの件数もカウントしてテーブルに追加することにした。(資料編：図7 [新着リクエスト一覧] 参照)

投票フォームについては、配置は変更せずリクエスト一覧の投票ボタンを押すと投票フォームが設けられたページへと移動することができる。(資料編：図10 [投票コメントフォーム] 参照)

【例】`$sql2 = "select * from web_clap where request_id=$request_id";`

`$data2 = mysql_query($sql2, $connect);`

`$count = mysql_num_rows($data2);`

上記の SQL 文では、投票コメントのデータ (投稿者. 日付. 本文など)

が保存されている web\_clap というデータベースのテーブルへアクセスし、request\_id と \$request\_id をリレーションすることで、どのリクエストに投票コメントが送信されたのかを結びつけて表示させている。その検索してきたデータを \$data2 の中に代入し、mysql\_num\_rows(\$data2) というコマンドによって \$data2 の中に入っている投票コメントデータをカウントして、さらに \$count という変数に代入している。この SQL 文の sql・data が sql2・data2 となっているのは、申請されたリクエストを検索している SQL 文との混同を避けるため、名称を変更しているからである。

この SQL を追加し投票コメント数をカウントしたことによって、一目で人気のリクエストを確認することができるようになった。(資料編：図 7 [新着リクエスト一覧] 参照)

sspj メニューの投票コメント管理メニューでは、リクエストに寄せられた投票コメントをすべて閲覧・編集することができ、選書システムを管理する図書館スタッフは本を選ぶ際に、学生からの意見をより反映しやすくなると考えられる。(資料編：図 14 [投票コメント管理] 参照)

投票管理メニューにはソート機能も実装したが、それは後の「ソートの実装と問題点」で説明する。

### (iii) 投票コメントの匿名性

ここまでは投票コメントの閲覧・投稿方法の変更について説明してきたが、次は投票コメントの投稿者について述べてこうと思う。

選書システムには引き継いだ時点から、拍手 (現在の投票コメント) にはコメント本文だけでなく、投稿者のニックネームも form で自動敵に送信されるようになっていた。利用者の登録情報は cookie に保存されているので、その情報を受け取れば投稿者のニックネームを表示する

ことが可能となっている。(資料編：図 11[投票コメント一覧] 参照)

ここで問題なのは、上記のような利用登録を行っていないユーザー(以下：非登録者)も、リクエストに対して投票コメントが送信できてしまうという点だ。利用登録者であれば自身のニックネームが表示されるが、非登録者は名前の欄が「guest」としてコメントすることができてしまう。これはコメントの匿名性を利用して、リクエストに対して誹謗中傷が書き込まれる可能性があるということになる。しかし、この問題は私だけでは判断することはできないので、後に記述するアンケートをお願いした際に図書館スタッフの方に意見を伺うことにした。

その結果、やはり匿名での投票コメント機能は利用不可にすべきとの意見だった。その理由として、先ほどの誹謗中傷が書き込まれるという可能性、そして「guest」として何件もの投票をされた場合、意見を反映させるのが難しくなるというものだった。

これらの意見から、非登録者の投票コメント機能は削除することを決めた。そのため、以前は同じファイルを使用していた TOP ページとマイページの「新着リクエスト一覧」を、別々のページに分断し新たにファイルを製作した。TOP ページのリクエスト一覧は if 文を使用し、投票があればクリックすることでコメントを閲覧することができ、それ以外の時は「コメントなし」と表示されるように場合分けを行った。(資料編：図 5 [新着リクエスト一覧] 参照)

#### (iv) E-mail 機能の検討

上記では管理者と学生との意見交換を行う機能として「投票コメント機能」について述べてきた。以前のリクエスト用紙にリクエスト内容を記入してもらう方式では、リクエストする学生の意見は記入できても、それに対する図書館スタッフの意見や回答を得ることはできなかった。

そのため学生の意見を反映できる参考にすることができる「投票コメント機能」は選書システムにおいて重要な機能だといえる。しかし、この機能も管理者と学生との直接的なやり取りをすることはできない。

そこで TOP メニューの「新規利用登録」から利用者登録をしてもらう際、この登録ページに E-mail フォームを設けることにより、管理者にメールを送信する機能を実装しようと考えた。(資料編：図 6 [新規利用登録] 参照) この案は昨年の卒業生も選書システムの中で導入を検討していたのだが、セキュリティ面で不安が残るため見送られていた。この E-mail 機能を今回実装してみようと考えたのだが以下の問題点が浮かび上がった。

1. 管理者である図書館スタッフの方々にメールのチェックをお願いしなければならない。
2. 個別に登録者管理メニューを作成しなければならない。
3. メール宛先を間違える可能性がある。
4. メールアドレスを変更した際、管理者へ報告しなければならない。

以上のような問題を製作中に解決することができなかつたため、今回も導入を見送ることにした。しかし、管理者と学生が密に意見交換できる機能は必ず必要となってくると思うので、今後選書システムを運用していく上での課題となってくると考えている。

#### **(5) ソートの実装と問題点**

実際に選書システムを実用化した場合、リクエストや投票コメントなどの送信数が増えて閲覧しにくくなってくる。そんな時に必要となってくるのがソート機能だ。このソート機能は引き継いだ際には実装されていなかった機能だが、私自身が作成時にリクエストリストの管理に苦勞

させられたため、実用化に伴い実装しようと考えた。

ソート機能を検討したページは以下のとおりである。

- ・ 新規リクエスト一覧 (TOP. マイページ)
- ・ 投票コメント管理 (sspj メニュー)
- ・ リクエスト一覧 (sspj メニュー)
- ・ 送信済みリクエストの状況確認

まずは投票コメント管理のソート機能から製作していった。ソートは3年次の授業で一度扱っており、それらのデータを参考にしながら取り掛かった。ソートする項目は日付・書名・著者名の3つで、それぞれラジオボタンで選択し、ソートボタンを押すことで実行される。(資料編：図14 [投票コメント管理] 参照)

SQL文には order by 句を \$joken という変数に代入し、

**【例】** \$sql2 = "select \* from web\_clap where request\_id=\$request\_id \$joken limit \$count, 20";

という SQL でデータを検索する。この文の中の変数 \$joken は (\$cmd == "sort") というコマンドが実行された際に

**【例】** \$joken = " order by \$sort desc";

という order by 句が実行され、\$sort にはソートしたい項目のレコードが代入される仕組みである。しかし、このプログラムで実際に実行はされるのだが、データベースにリクエスト情報を保存する際に、署名や著者名に読み仮名を記入するフォームがなかったため、読み仮名が保存されていない状態となっていた。そのため、通常のソートのような「あいいうえお順」といった並べ替えは不可能となってしまった。読み仮名の記入フォームを追加することはできるのだが、これ以上記入項目を増やすことは利用者の負担の増加に繋がると判断したため、変更は加えなかつ

た。

そしてもう一つ、上記のデータベースの情報不足以外に、登録されたデータが混在しているため SQL が不具合を起こしているという問題も発覚した。これはデータベースのテーブル内でのデータの登録ミスが原因で、このエラーのために以下のページでソート機能の実装を見送ることに決めた。

- ・ 新規リクエスト一覧 (TOP. マイページ)
- ・ 投票コメント管理 (sspj メニュー)
- ・ リクエスト一覧 (sspj メニュー)
- ・ 送信済みリクエストの状況確認

結果、不完全な状態ではあるが、投票コメント管理ページにのみソート機能を付け加えることにした。これらの問題点は、今回の選書システムの反省点の一つといえる。

## (6) 一括編集・削除機能

次は新たに導入したチェック機能を用いた一括編集 (受付状況)・削除機能について説明していきたい。sspj メニューではリクエストや投票コメントの編集と削除を行うことができる。以前の選書システムでは、リクエストや投票コメントは一件ずつ削除されていたが、システムが実用化され、多くの情報が送信されてきた場合に対処しきれなくなると予想されたため、一括で作業できる機能を導入した。(資料編：図 14 [投票コメント管理] 参照)

まずはチェック機能の作成だが、リクエスト一覧のテーブルにチェックボックスを追加していくところから始める。

【例】 `input type="checkbox" name="chbox[]" value="$request_id"`

次に、自分で好きな関数を定義することができる function を用いて「function checkout()」「function checkon()」という二つの関数を head 内で定義する。

【例】function checkon()

```
function checkon() {  
    var kensu = document.FORM.elements.length;  
    for (i=1; i <= kensu; i++) {  
        if (document.FORM.elements[i].type == "checkbox") {  
            document.FORM.elements[i].checked = true;  
        }  
    }  
}
```

そして、全てのチェックボックスに結果を反映させる以下のボタンを作る。(資料編：図 14 [投票コメント管理] 参照)

【例】input type="button" value="全てにチェック" onClick="checkon()"  
input type="button" value="チェックを外す" onClick="checkout()"

これらのボタンを押すと、さきほど定義した関数が実行される仕組みとなっている。あとは編集する受付状況 (受付中. 検討中. 購入予定. 保留. 購入済み. その他) をプルダウンメニューで作成し、koshin というコマンドを hidden で更新ページに飛ばせば完了となる。これらと同じ方法で投票コメントの一括削除も作成し、実装することができた。

## 4 評価

### (1) アンケート結果

今回製作した選書システムをゼミの 3 回生と図書館スタッフ・選書委員の方々に使用してもらい、アンケートに答えてもらった。その結果次のような結果となった。

(i) 3 回生の意見

1. デザイン：デザインは引き継ぎ後にメニューなどの再配置を行ったため、高評価を得ることができた。
2. 見やすさ：とても良いという評価が多かったが、横幅が広くスクロールしなければ見えにくいとの意見もあった。
3. 色合い：色合いも引き継ぎ前はグレー中心のデザインだったが、明るい水色に変更し、綺麗でさわやかな色合いという評価を得た。
4. 使いやすさ：悪くない評価だったが、ご利用案内をもう少し分かりやすい位置に表示してほしいなどの要望もあった。
5. メニューの見たい目：メニューの h3 タグに CSS で装飾を施したことで見やすいデザインになったが、項目が少し多く分かりにくいという意見があった。
6. 機能面：機能面はとても充実しているという声が多く聞かれた。しかし、「戻る」ボタンの有無やメニューの項目が専門的過ぎて分かり辛いなどの意見もあった。

3 回生には新規利用登録からリクエストの申請までの、主に学生が利用する機能を中心に使用してもらいアンケートに答えてもらった。その結果、上記のような結果を得ることができた。全体的に見ると高評価を得ることができた項目が多かったように思えるが、一方で選書システムの利用者でもある学生の立場から意見を聞いたことで、今まで気づかなかった問題点も多数浮かび上がってきた。例えば、ページ全体の横幅が広いと見えにくい。「ご利用案内」内にリンクを作成し新規登録画面に飛ばしたほうが良いなど、初めて利用する学生の目線から多くの意見を頂き、レイアウトを改善していくことができた。ただβテスト時に投票コメント機能やリクエスト検索など、いくつかの機能について時

間の関係上、詳しく説明できなかったことが今回の反省点である。しかし、これらの機能については図書館スタッフの方に意見を聞くことができたので次項で説明していきたい。

## (ii) 図書館スタッフ・選書委員の意見

図書館スタッフの方々には、2時間ほどお時間を頂けたので、選書システムの機能を全体的に説明することができた。一つひとつの機能に対し時間をかけて説明することができたので、管理者の立場から貴重な意見を多数聞くことができた。

まずは TOP ページだが、以前はページを開くと管理者からのお知らせが表示されるようになっていた。しかし、初めて利用する人のためにご利用案内を表示すべきとの意見が挙がった。TOP ページには最新のお知らせが表示されたほうが、図書館スタッフからの連絡事項に目を通すことができ良いのではないかと感じていたが、3 回生に行った β テスト時にも数人から同じ点を指摘されたので、ご利用案内を最初に表示するよう修正することにした。(資料編：図 2 [TOP ページ] 参照) さらに、「拍手機能と投稿コメント」の「投票コメントの匿名性」でも述べた guest(利用者登録を行っていないユーザー) での投票コメント機能も、誹謗中傷の書き込みがされる可能性があることから廃止とした。これにより、「投票コメント機能」は利用者登録を行った学生のみが利用できる機能となった。「書籍データ詳細」で確認できる備考欄(リクエスト申請する際に理由などを書き込むフォーム)については、すべてが利用者に見せてよい内容ではないため、管理者のみが閲覧できるように変更を加えてほしいという要望をもらった。これについては、mypage\_request\_detail.php 内の備考欄を削除することで要望にこたえることができた。(資料編：図 10 [投票コメントフォーム] 参照) その他、

ログイン方法の見直しなどの意見も頂いたが、これは後の「改善点と今後の課題」で説明したいと思う。

次に管理ページについての意見だが、「リクエスト申請の際に Amazon のリンクも送信してほしい」「リクエストは一人 5 件までに制限してほしい」「発注作業も兼ねることはできないか」「登録者の管理ページがほしい」などこれまで学生からのリクエストを管理してきた図書館スタッフならではの貴重な意見が多く聞かれた。

## (2) 改善点と今後の課題

このβテストとアンケートで3回生と図書館スタッフ・選書委員、この3つのグループから意見を聞くことができたが、フレームの幅の問題やご利用案内の配置変更など、3回生からの要望についてはほぼ改善することができた。しかし、図書館スタッフ・選書委員の方からの意見は一筋縄ではいかない要望が多く、ごく一部しか改善することができなかった。

まずリクエスト管理と兼ね合わせて発注作業も行いたいという意見だが、このシステムは学生から気軽にリクエストを受け付けられるよう開発されたシステムのため、趣旨から外れるため実装は見送った。次にマイページにログインする際の問題だが、現在は学生番号(7桁の数字であれば良い)とパスワードで利用者登録できるよう製作されている。(資料編：図2 [TOP ページ] 参照) この仕様については、学生番号が重複しても登録できてしまう点や、7桁の番号であれば学生番号以外でも登録できてしまう、マイページの退会手続きがないなど、セキュリティ面を不安視する意見が多く聞かれた。さらに、sspjメニューにログインするためのパスワードも、変更するにはプログラムを直に書き直さなければ

いけないため、定期的に変更することが難しいのではないかといった意見も頂いた。しかし、これらの問題はログイン方式を根本的に見直す必要があり、技術的にも時間的にも改善不可能な問題となってしまった。

その他の Amazon のリンクをリクエストデータと一緒に送信するという機能は、データベースに新たなレコードを追加しなければいけないが、これについては時間が許す限り実装を目指して製作していきたい。(資料編：図 8 [リクエストの申請] 参照)

このような多くの問題を抱えていることが分かった選書システムだが、嬉しいことに実際に図書館で実用化してもらえることが決まった。実用化されることを第一に考え製作に取り掛かってきたので、目標が一つ達成されることとなったが、まだまだ改善点は残っているので、少しでも使いやすくなるよう努力していきたいと思う。

## 5 結論

今回この学生選書システムを引き継いだ事により、php に対する理解が深まっただけでなく、「人の役に立つものを作る」というゼミのテーマに沿った卒業制作ができたのではないかと思う。

製作を行っていく中で実装を見送った機能や、アンケートの意見に答えることができなかった機能も多くあった。これらを導入できなかったことに少し悔いが残る。しかし、製作当初から目標としていたシステム実用化の目処も立ち、多くの学生が自分の作ったシステムを利用してくれると思うと、とても嬉しく思う。

この学生選書システムは、多くの人の意見を取り入れながら改善を重ねてひとまず完成を向かえることができた。そんな中、物作りとは第三者の視点を大切にしながら、システムを使用する人のことを一番に考え

——学生選書システムの実用化にむけて——

ることが大事なのだと知ることができた。これが今回私が卒業制作によって学んだ一番の収穫であるように思う。

**注**

- (1) 「学生選書支援オンラインシステムの構築について」(2009 年度卒業論文、影林泰佳さん)
- (2) <http://www.amazon.co.jp/>
- (3) <https://affiliate.amazon.co.jp/gp/advertising/api/detail/main.html>

**文献表**

とほほの WWW 入門：<http://www.tohoho-web.com/>