愛知工業大学研究報告
第 46 号　平成 23 年

携帯端末用個人向け書籍管理システム

A personal book management system on mobile phone

久保田 辰也†,　沢田 克敏†
Tatsuya Kubota,　Katsutoshi Sawada

Abstract　A personal book management system which works on mobile phone was developed. This paper describes the construction and the function of this system. It also describes security techniques used in this system. This system employs PHP for web server and MySQL for database server. Cooperating with "Amazon Product Advertising API", this system can provide a lot of book information. Only by inputting the ISBN of the book, users can easily obtain the title, author and more of the book. Anyone can try this system freely by accessing to "http://m.orehon.com/".

1．はじめに

新古書店を日々利用していると、書籍の所持数は増加していくばかりであり、自分の書籍の所有状況や書籍内容が十分に整理・把握できなくなってしまうという問題が悩みの一つである。そこで、書籍の所有状況を管理していきたいと考え、携帯端末を用いた個人向け書籍管理システムを構築した。

本論文では、構築したシステムの構成と動作を設計の考え方を交えて詳細に述べる。また、本システムの構築においてセキュリティに関して特に注意を払った点についても述べる。

2．システムの構成

本章では、今回作成したシステムの構成について実際の動作などを交えて説明する。

2・1　構成概要

本システムの構成概要を図 1 に示す。

本システムは PHP[1)]を用いて記述されており、データベースサーバには MySQL を採用している。また、アマゾンジャパン株式会社が提供する Product Advertising API[2)] (以下 Amazon API)を利用することにより、Amazon.co.jp 内に存在する膨大な商品情報との連携を行っている。

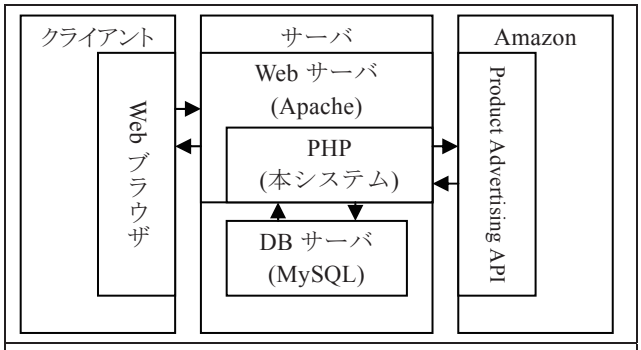

図 1 構成概要

2・2　共通処理

本システムでは基本的に毎回 index.php へアクセスする設計となっている。つまり、状況に応じて index.php から各機能を呼び出していることになる。このような手段をとることで、毎回実行する必要がある共通処理(例えばセッション情報の確認等)が確実に実行される。

以下の各節で実際の処理内容を述べる。

2．2．1　PHP の環境設定

エラー報告レベル、エラーログ保存先、内部エンコーディングなどの設定を行っている。ただし、PHP ソースから変更できない項目については、別途 .htaccess ファイル内で設定している。

2．2．2　本システムの設定の読み込み

システムタイトル、データベース情報、Amazon API アクセスキーなどを settings.php から読み込んでいる。サーバや運営者によって変化する情報は、設定ファイルという形にまとめておくことで管理しやすくなる。

†　愛知工業大学 工学部 電気学科 (豊田市)

2．2．3　ユーザ定義関数の宣言

使用頻度の高いユーザ定義関数をあらかじめ宣言しておく。共通処理の段階で宣言を行っておくことにより、個別の PHP ソース内で宣言する必要が無くなる。

2．2．4　セッション情報の確認

セキュリティを確保するため、UserAgent (ブラウザ情報)の照合を行う。ログインが完了した後に、前回アクセス時と UserAgent が変化した場合は、強制的にログアウトさせる。

2．2．5　文字コードの調整

UserAgent を確認し、ブラウザへ出力すべき文字コードを決定する。続いて、ユーザからの入力情報を内部処理用の文字コードに合わせるため、UTF-8 形式へ変換する。また、本システムではブラウザへの出力情報を一旦全て変数内へ格納している。全ての出力情報が準備できた段階で、変数の内容を最初に決定した文字コードへ変換し、ブラウザへ出力している。

2・3　新規登録

新規登録ページは図 2 のように表示される。

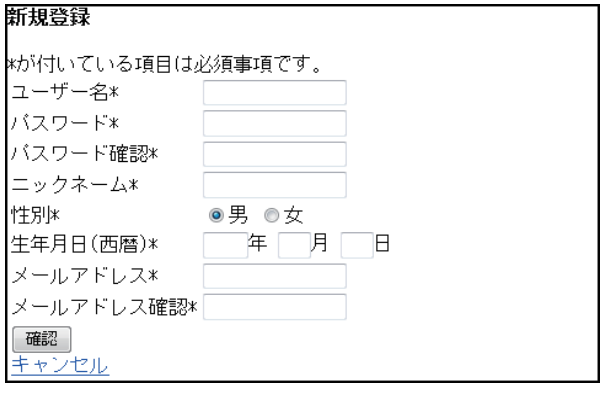

図 2 新規登録ページ

図 2 のページの空欄を不備無く補充し、"確認"を押すと次の図 3 のようになる。なお、図 3 に含まれる少々読みにくい文字列は CAPTCHA と呼ばれている。この文字列が正しく判読されているかを確認することで、機械的な多重登録を防ぐことが可能となる。なお、この文字列を含む画像の生成には kcaptcha[3]を利用している。

CAPTCHA 認証に成功するとユーザ情報がデータベースへ登録され、ユーザは登録した情報を元にログインが可能となる。

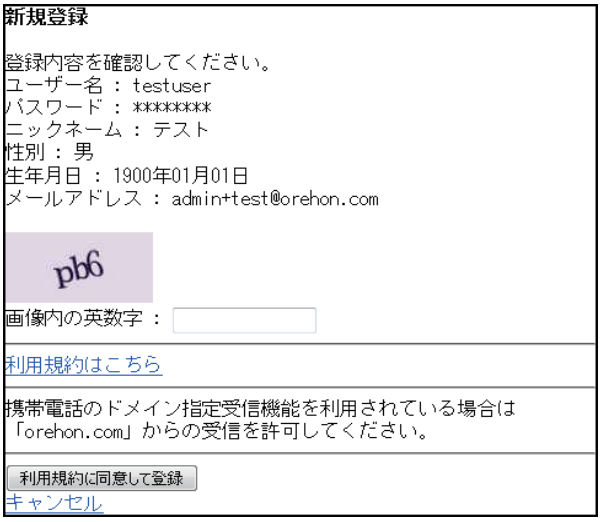

図 3 新規登録確認ページ

また、登録が完了した時点で、表 1 のようなメールがユーザへ送信される。

表 1 新規登録メール (抜粋)

件名	オレほん(β) - メールアドレス認証
本文抜粋	以下の URL を開いて頂きますと、メールアドレスの確認が完了致します。 http://m.orehon.com/index.php?mode=mailauth&idstr=testuser&authstr=[32 桁の英数字]

ユーザが表 1 内の URL へアクセスすると、図 4 のように登録されたメールアドレスが正確であると判定される。

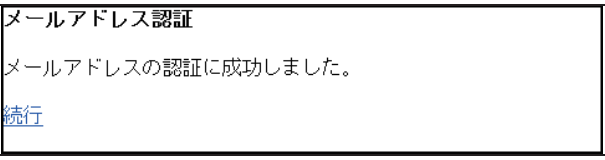

図 4 メールアドレス認証ページ

しかし、新規登録から 72 時間以内にこの操作を行わなかった場合は、アカウントが停止されユーザはログインが不可能となる。ただ、ユーザがメールアドレスを誤って登録してしまう可能性も考えられる。そういった場合は、アカウントが停止される前に登録情報の編集を行えば、再認証を受けることが可能となっている。

また、ユーザが ID やパスワードを亡失した場合に備え、ユーザ自身が照会できる手段を用意しておく必要がある。本システムでは、新規登録時に設定された誕生日とメールアドレスを用いて本人確認を行う。

2・4　ログインとログアウト

2．4．1　概要

ここでは図 5 のログインページについて説明する。ログイン方法にはユーザ名とパスワードを用いる通常ログインと、携帯端末の固有情報を用いた簡単ログインがある。簡単ログインボタンは、携帯端末からのアクセスだと判断される場合のみ表示される。なお、補足としてゲストログインについても紹介する。また、ログインした段階でメールアドレスの認証が完了していない場合は警告が表示される。認証の有効期限内であればそのまま操作を継続できるが、有効期限を過ぎている場合は強制ログアウトとなる。

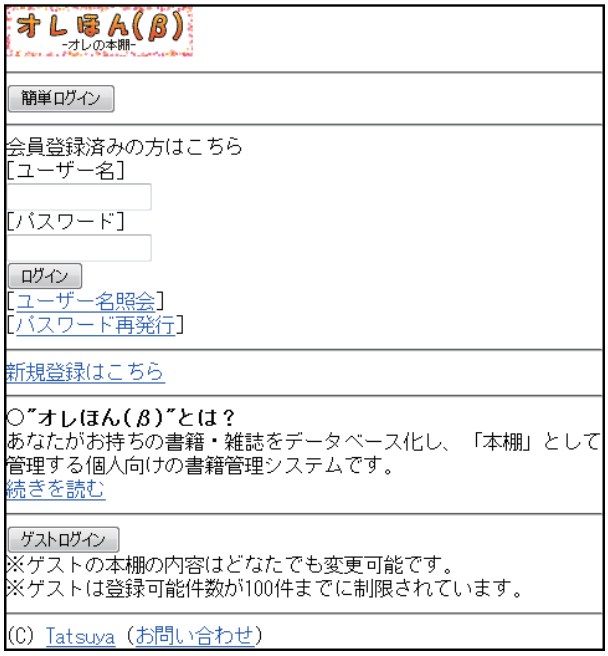

図 5 ログインページ

2．4．2　通常ログイン

新規登録時に作成したユーザ名とパスワードを送信することでログインできる。一般的なログイン方法である。

2．4．3　簡単ログイン

携帯端末の回線固有情報 [4]などを用いることで、ログインを可能とする。ユーザ名やパスワードの入力を省き、1 ボタンでログインできる。ただし、新規登録後は一旦通常ログインを行い、メニューから図 6 のような簡単ログイン設定を実行する必要がある。

なお、簡単ログイン機能はセキュリティ低下の要因となりやすいが、本システムにおけるセキュリティ対策については 3 章で述べる。

図 6 簡単ログイン設定ページ

2．4．4　ゲストログイン

誰でも試用できるよう、ユーザ名"guestuser"・パスワード"guestpass"としてログインする機能である。このため、通常ログイン用の入力欄へ同一の文字列を入力することでもゲストログインは可能である。なお、ゲストログイン時に適用される制限は、データベース内のユーザ管理テーブルを直接編集することで設定できる。

以上のいずれかの方法でログインに成功すると、図 7 のようなメニューページが表示される。

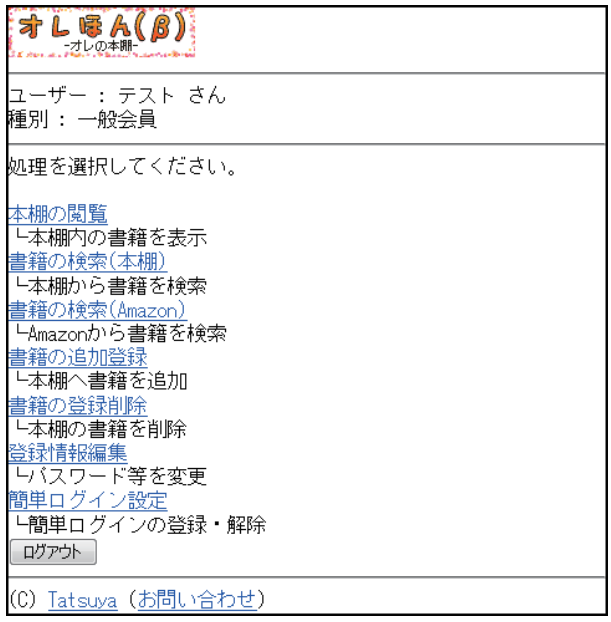

図 7 メニューページ

2．4．5　ログアウト

メニューからログアウトを実行すると、サーバに保存されているセッション情報を破棄してログアウトが完了する。

2．4．6　管理者への問い合わせ

ログインページ(図 5)とメニューページ(図 7)には管理者への問い合わせ用のメールフォームが用意されている。入力内容の確認後にユーザが送信内容の確定を行うと、管理者宛にメールが送信される。なお、送信されたメールの末尾には送信を行ったユーザの情報が含まれているため、問題解決のヒントとなる。

2・5　登録情報編集

登録情報編集ページは図 8 の通りである。このページからパスワードやニックネーム、メールアドレス、本棚(書籍リスト)の公開設定の変更が可能である。なお、メールアドレスを変更する場合に限り、新規登録時と同様に CAPTCHA とメールアドレスの認証が必要となる。

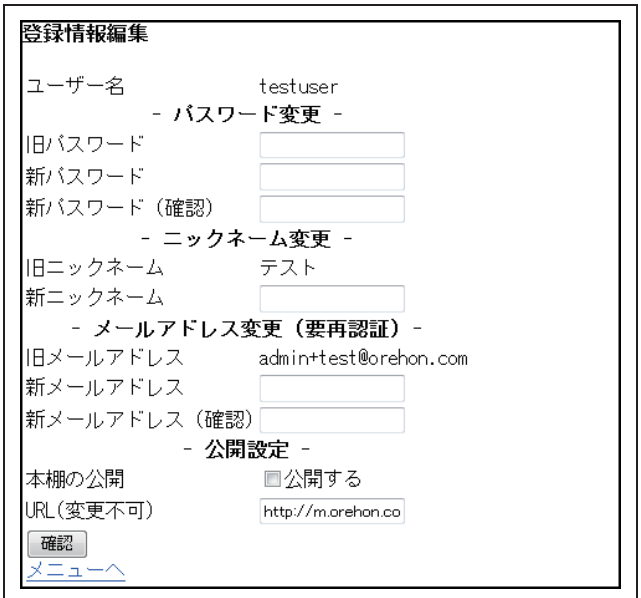

図 8 登録情報編集ページ

2・6　書籍管理

本システムの中核をなす書籍管理について紹介する。ここではユーザが所持している書籍の情報を"本棚"と呼ぶことにする。

2．6．1　書籍の登録

書籍の登録には ISBN[5]を用いる。ISBN とは書籍一冊ずつに割り当てられている番号である。この番号を本システムへ入力することにより、Amazon API との連携が可能となる。図 9 にその例を示す。

入力された ISBN について検査を行い、正しくない番号であれば警告を表示する。なお、10 桁の ISBN は末尾が"X"となる場合があるが、入力の際には手間となる。そこで、"X"を"0"と置き換えて入力しても正しく処理できるよう構成した。続いて、入力された ISBN を元にユーザの本棚を検索し、登録済みの書籍は重複登録されないようにする。ユーザの本棚に無い場合は Amazon API を用いて書籍情報を取得し、ユーザの本棚へ追加する。

2．6．2　書籍の削除

ページの構成自体は図 9 とほぼ同じである。書籍の追加登録の場合と同様に ISBN を検査し、正しい場合はユーザの本棚を検索して、該当する書籍があれば削除する。

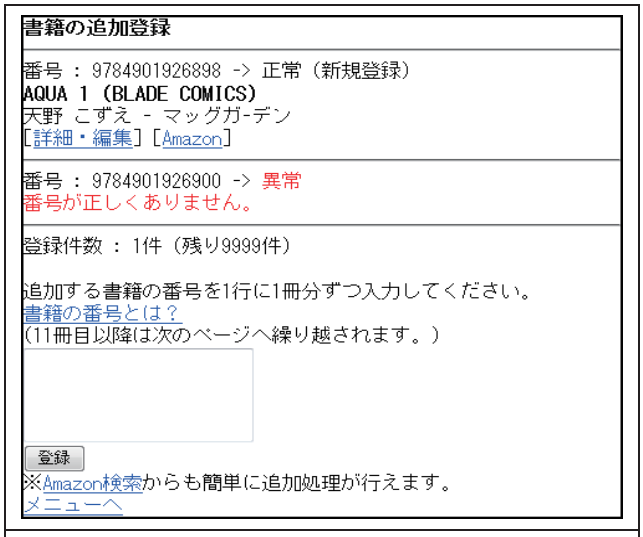

図 9 書籍の追加登録ページ

2．6．3　本棚の閲覧・検索・公開

ユーザの本棚に登録されている書籍を検索・表示する。検索条件を指定しなかった場合は、全ての書籍が表示される。なお、他者へ本棚を公開する場合も同じ仕組みを用いている。本棚を閲覧する場合の表示を図 10 に示す。

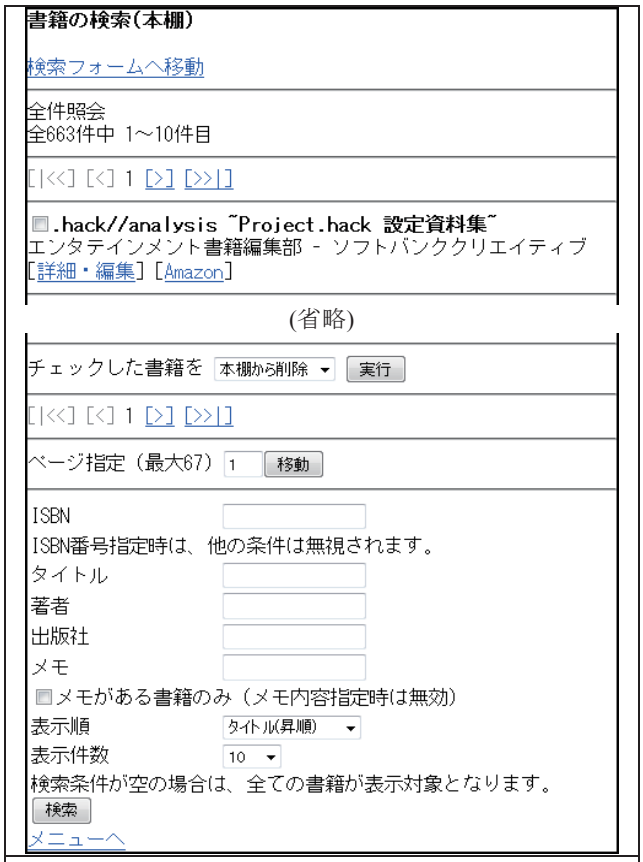

図 10 本棚の閲覧ページ (抜粋)

図 10 のページ下部に設置されたフォームから検索条件を指定できる。ただし、検索条件を指定せず表示順・

表示件数のみを変更し、改めて表示することも可能である。また、書名の前にあるチェックボックスを利用することにより、複数の書籍を本棚から一括で削除できる。

2．6．4　書籍の検索

Amazon.co.jp 内を検索する場合は、フリーキーワード検索もしくは詳細検索(図 11)が選択できる。なお、実際にフリーキーワード検索を行うと図 12 のように結果が表示される。

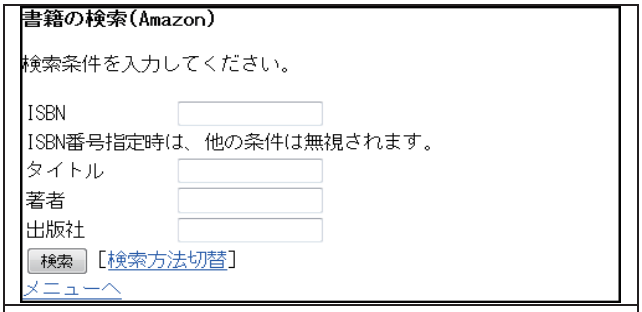

図 11　詳細検索ページ

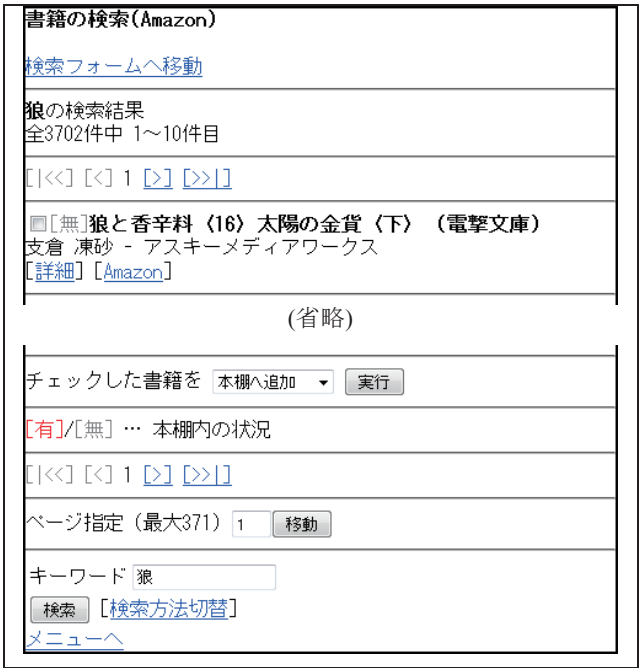

図 12　フリーキーワード検索結果ページ

図 12 には本棚を検索した場合と同じように、書名の前にチェックボックスがある。しかし、このページの場合は書籍の削除だけでなく、チェックした書籍を本棚へ追加する機能も持っている。また、ユーザの本棚内にその書籍が存在しているかを"[有]" もしくは "[無]"として表示するため、一目で判別することができる。

2．6．5　書籍の詳細情報

各ページで表示される "[詳細]" を開くことで、書籍の詳細な情報が閲覧できる。例として図 13 に単行本 [6]の詳細情報ページを掲載する。

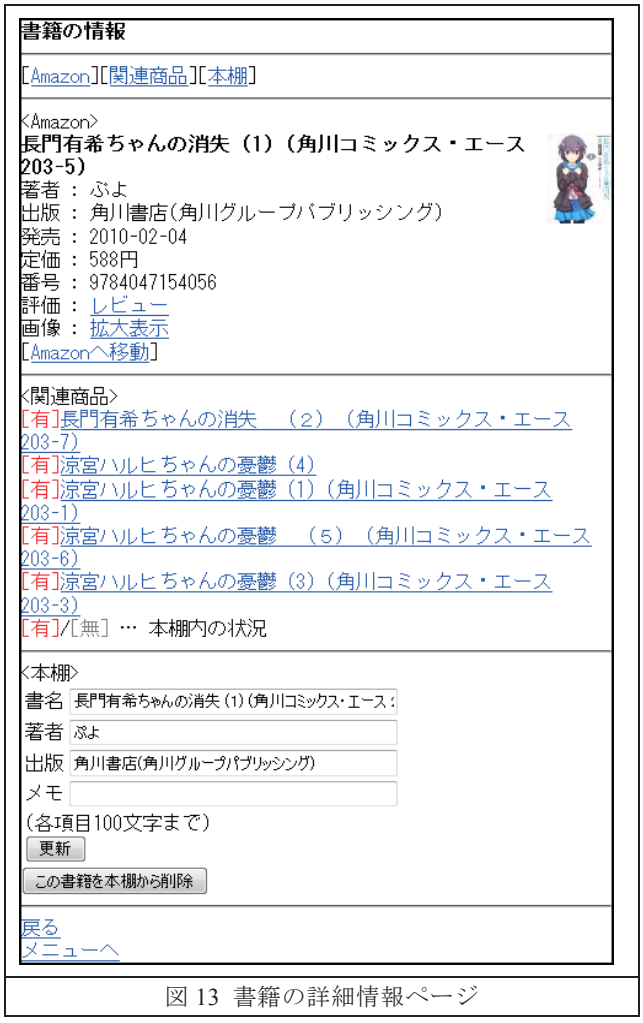

図 13　書籍の詳細情報ページ

図 13 にあるように、Amazon API より取得した情報がページ上部へ表示される。また、関連商品の部分はユーザの所有状況と比較できるようになっている。なお、ページ下部にあるユーザの本棚内の情報は、ユーザ自身が自由に編集可能である。

2．6．6　レビュー閲覧

レビューの表示方法について説明する。まず、Amazon API からはレビューの文章自体ではなく、埋め込み用ページの URL しか提供されない。従って、PC からのアクセスの場合は<iframe>タグが利用できるので、本システム上でページを埋め込んで表示することが可能となる。しかし、携帯端末からのアクセスの場合は埋め込みページに対応していないので、Amazon.co.jp 自体のレビューページへ転送処理を行うことで対応している。

3．セキュリティ

本システムの構築にあたって、セキュリティの観点から注意を払った点について述べる。

3・1　クロスサイトスクリプティング対策

3．1．1　概要

クロスサイトスクリプティング [7]とは、ユーザが入力した文字列を、無害化せずそのまま出力することで発生する脆弱性のことである。では、攻撃者が悪意のあるHTML や JavaScript を含む文字列を準備し、サーバへ送信した場合について考えてみたい。例えば、送信された情報が保存・表示される掲示板などの Web サイトの場合を想定してみる。さて、攻撃者が送信した HTML などがそのまま出力されてしまった場合はどうなるだろうか。一般の閲覧者のブラウザ上で当該 HTML などが実行されてしまい、被害につながる可能性がある。具体的には危険な Web ページを強制的に表示したり、パスワードなどを攻撃者へ送信したりするといった問題が考えられる。

3．1．2　対策

対策はさほど難しくなく、HTML タグの一部として利用される< > " &の無害化を行えばよい。具体的には、送信された文字列に対して、以下のような置換処理を行う。

・< → <　・> → >
・" → "　・& → &

また、PHP には専用の関数である htmlspecialchars()が準備されているので、この関数を利用すれば手軽である。

では、攻撃者から以下の文字列が送信された場合を例として考えてみたい。

・送信された文字列:
秘密

この文字列をそのまま出力した場合、ブラウザではどのように処理されるだろうか。

・HTML ソース:
秘密
・ブラウザ上での実際の表示:
秘密

攻撃者が意図した通り、危険なページへのリンクが表示されてしまった。この状態は大変危険である。

続いて、htmlspecialchars()を通した場合を見てみる。

・HTML ソース:
<a href="http://www.example.com/virus.html
">秘密
・ブラウザ上での実際の表示:
秘密

このように、攻撃者が意図した通りのリンク表示にはならず、攻撃を無力化することができる。

3・2　SQL インジェクション対策

3．2．1　概要

SQL インジェクション [8]とは、データベース接続時に使用する SQL 文への攻撃のことである。例えば、書籍管理テーブル"books"を書名によって検索する場合について考えてみる。

・ユーザが"テスト"と入力した場合に実行される SQL 文
SELECT * FROM ‘books’ WHERE title = ‘テスト’;
・ユーザが"テスト’; SELECT * FROM ‘users"と入力した場合に実行される SQL 文
SELECT * FROM ‘books’ WHERE title = ‘テスト’; SELECT * FROM ‘users’;

1 つめは通常の SQL 文となり正しく検索される。しかし、2 つめは”users”というテーブルが存在した場合に、その中身が全て表示されてしまう危険性がある。このように、検索クエリなどに細工を施すことによって、攻撃が行われる場合がある。

3．2．2　対策

対策としてプリペアードステートメントという方式を採用している。この方式は SQL 文内の可変箇所をあらかじめ定義しておく手法だ。こうすることで、SQL 文内の可変箇所内に SQL 制御語句が含まれてしまった場合でも、自動的に無視されるようになる。具体的には以下のような SQL 文を宣言しておく。この場合、後から専用の関数を用いて、":TITLE"部分へ検索文字列を当て込む。

・SELECT * FROM ‘books’ WHERE title = :TITLE;

3・3　パスワード管理におけるセキュリティ

3．3．1　概要

複数の Web サイトで同じパスワードを使用するユーザも多いので、パスワードの流出対策は重要である。万一、データベースの内容が漏れてしまった場合でもパスワードが流出しないよう、対策を講じておくべきである。

3．3．2　対策

例えば、ユーザが"mypassword"というパスワードを登録したいとする。まず、その文字列を md5()というハッシュ関数へ送る。すると、以下のハッシュ値が得られる。

・34819d7beeabb9260a5c854bc85b3e44

そして、このハッシュ値をデータベースへ登録する。ハッシュ値からパスワードを復元することは困難なので、流出した場合でも安全性が保たれる。また、同じパスワードからは同じハッシュ値が生成されるため、ログイン時にはこのハッシュ値同士を照合することで認証が可能となる。ただ、上記の通りデータベース上に元のパスワ

ードを保持していないので、ユーザがパスワードを亡失した場合は、システム側で再発行を行う必要がある。

３．４　簡単ログインに関するセキュリティ

３．４．１　概要

簡単ログインは携帯端末の固有情報を用いて認証を行うが、PC からアクセスした場合には偽装可能な項目も多い。確認項目を増やし、不正なログインからシステムを守る必要がある。

３．４．２　対策

本システムでは簡単ログインを行う際に 4 つの情報を照合し、安全性を高めている。

・回線契約ごとのユーザ識別子 (UID)
・携帯端末の機種名 (UserAgent)
・本システムが個別に発行する固有情報 (Cookie)
・携帯キャリアの回線 (IP アドレス帯 [9) 10) 11)])

IP アドレスは偽装が難しく、対策の有効性が高い。

３・５　セッションハイジャック対策

３．５．１　概要

セッションハイジャック [12)]とは、サーバがユーザを識別する際に用いる文字列(セッション ID)を盗用もしくは偽装し、他のユーザになりすます行為のことを指す。セッション ID は基本的に Cookie を用いて管理されるが、古い携帯端末だと Cookie に対応していない場合がある。そのような場合は、以下のように URL の末尾にセッション ID を付加して利用する。

・http://m.orehon.com/?OHSESSID=[ランダムな英数字]

しかし、URL に付加した場合はリンク元情報(Referrer)からセッション ID が漏れやすい。セッション ID が漏れたことを検出し、強制的にログアウトさせるような仕組みが必要である。

３．５．２　対策

まず、外部サイトへアクセスする際には、URL にセッション ID を含まないリダイレクトページを挟むこととした。これで Referrer 問題に対応できる。また、セッション ID が漏れてしまった場合は、端末情報(UserAgent)を用いて判断する。ログイン時と異なる UserAgent でアクセスした場合には強制的にログアウトさせ安全を保つ。

４．むすび

増え続ける書籍の管理の効率化を図ることを目的として、携帯端末を用いた個人向け書籍管理システムの構築を行った。本論文では、構築したシステムの構成と動作を設計の考え方を交えて詳細に説明した。また、本システムの構築においてセキュリティに関して特に注意を払った点についても述べた。

本システムは Amazon.co.jp 内の商品データベースを利用することができ、利便性も非常に大きい。最終的に個人向け携帯端末用書籍管理システムが完成したことにより、書籍管理にかかる手間や時間が大幅に短縮された。今後、実際に使用した結果を反映させて、本システムの機能改善を行う予定である。

なお、本システムは Web 上で公開を行っており、ユーザ登録を行えば誰でも利用可能となっている。URL は次の通りである。　　http://m.orehon.com/

参考文献

1) PHP: PHP マニュアル - Manual
(2011 年 3 月 15 日閲覧)
http://www.php.net/manual/ja/
2) Product Advertising API (2011 年 3 月 15 日閲覧)
https://affiliate.amazon.co.jp/gp/advertising/api/detail/main.html
3) KCAPTCHA project - CAPTCHA.ru
(2011 年 3 月 15 日閲覧)
http://www.captcha.ru/en/kcaptcha/
4) ケータイの端末 ID・ユーザ ID の取得についてまとめてみました (2011 年 3 月 15 日閲覧)
http://ke-tai.org/blog/2008/09/08/phoneid/
5) ISBN (2011 年 3 月 15 日閲覧)
http://www.infonet.co.jp/ueyama/ip/glossary/isbn.html
6) ぷよ: 長門有希ちゃんの消失 (1), 角川書店, 2010.
7) @IT：クロスサイトスクリプティング対策の基本
(2011 年 3 月 15 日閲覧)
http://www.atmarkit.co.jp/fsecurity/special/30xss/xss01.html
8) 今夜分かる SQL インジェクション対策 － @IT
(2011 年 3 月 15 日閲覧)
http://www.atmarkit.co.jp/fsecurity/column/ueno/42.html
9) 作ろう i モードコンテンツ | サービス・機能 | NTT ドコモ (2011 年 3 月 15 日閲覧)
http://www.nttdocomo.co.jp/service/imode/make/
10) KDDI au: EZfactory (2011 年 3 月 15 日閲覧)
http://www.au.kddi.com/ezfactory/index.html
11) MOBILE CREATION (2011 年 3 月 15 日閲覧)
http://creation.mb.softbank.jp/
12) @IT：Web アプリケーションに潜むセキュリティホール（3） (2011 年 3 月 15 日閲覧)
http://www.atmarkit.co.jp/fsecurity/rensai/webhole03/webhole01.html

(受理　平成 23 年 3 月 19 日)