

2012年度卒業研究

インターネットショッピングにおける  
クレジットカード利用補助システムの検討

指導教員 渡辺恭人

政策情報学部4年

学籍番号 0940186

松本 拓也

## 概要

インターネットの普及によりできることが増えてきた今日、インターネットショッピングは家にいながら買い物をすることが出来る手段として普及している。配達時間などのさまざまなオプションの他、代金の支払方法も自分で好きなものを選べるような形を取れるようになっているショッピングサイトがほとんどになり、その中でコンビニ支払い、代金引換支払い、クレジットカード支払いの三種類が主な支払方法であるが、クレジットカード支払いを利用した際、すぐには口座から代金が引き落とされず、支払日にその月利用した分の利用額がまとめて引き落とされる形になっている。

学生でもクレジットカードを持つことができるが、社会人と違い使える額の制限が大きいので使いすぎに気をつけなければならないため、カード会社のマイページなどで確認するだけでなく手元で管理を行なえるシステムの作成を行った。

目次	
概要	2
目次	3
1. 背景と目的	5
1.1 背景	5
1.2 目的	5
1.3 本論文の構成	5
2. 現状と問題点	6
2.1 ショッピングサイト	6
2.1.1 ショッピングサイトの使い方	7
2.2 ショッピングサービスの支払方法	7
2.2.1 代金引換	8
2.2.2 コンビニ支払い	8
2.2.3 クレジットカード支払い	8
2.3 大学生の平均生活費	9
2.4 問題点	9
2.5 現在あるサービスと対策	10
3. 解決への検討	19
3.1 解決すべき問題	19
3.2 解決へのアプローチ	19
4. システムの設計と実装	20
4.1 システム設計	20
4.2 データベースの設計	21
4.2.1 アイテムコード保存用データベース「itemcode」	21
4.2.2 計算用数値保存用データベース	22
4.3 システムの実装	22
4.3.1 検索機能	23
4.3.2 登録	29
4.3.3 ほしいものリスト、計算システム	31
4.3.3.1 ほしいものリストと計算機能	34

4.3.3.2	計算結果反映	45
4.3.3.3	商品の削除	47
4.3.3.4	残額リセット	54
5.	評価と考察	56
5.1	評価	56
5.2	考察	57
6.	まとめと今後の課題	59
6.1	まとめ	59
6.2	今後の課題	59
	参考文献	61
	謝辞	62

## 1. 背景と目的

### 1.1 背景

インターネットショッピングサイトは、amazon.co.jp、楽天市場などがあり、これらのショッピングサイトでは、支払い方法の種類として、クレジットカード支払い、代金引換支払い、コンビニ支払いがある。この中で最も簡単で手軽な支払い方法はクレジットカード支払いである。しかし、代金引換やコンビニ支払いとは違い、自分の財布から支払いをするのではなく、支払日にそれまで使った金額がまとめて登録した口座から引き落とされるため、自分の思っている金額以上にお金を使ってしまう危険性がある。特に学生がクレジットカードをインターネットショッピングで利用する際には個人の状況にもよるがやはり使える額が社会人と比べ少ないため、確認することだけでなく管理をしやすくできる必要がある。

### 1.2 目的

本研究では特に学生など使える額が大きく制限されてしまう状況にある人がクレジットカードを利用するにあたり、クレジットカード会社のサイトを利用しての利用額確認ではなく、自らほしい物と月の利用額の把握を気軽にできるようにし、金銭管理の意識を高めていけるようにすることを目的とする。

### 1.3 本論文の構成

本論文は、2章では現状と問題点を洗い出し、現状ある対策やサービスなどを分析、考察する。3章ではそれを踏まえたうえでの解決方法の提案、実装の方法について述べる。

4章ではネットショッピングの支援システムの詳細設計と実装を行い、5章で評価、6章で本研究のまとめを行う。

## 2. 現状と問題点

ここではショッピングサイトの支払い形態やそのメリットやデメリットについて議論していく。

### 2.1 ショッピングサイト



図 2.1-1 ショッピングサイト amazon.co.jp

インターネットショッピングサイトとは、インターネットを通じて買い物を行う通信販売サイトである。利点として家にいながら買い物ができる、店舗より安く出品されているものが多々ある、気軽に地方の特産品などを取り寄せることができるなどが挙げられる。その他にも商品の購入者などが商品の評価を書き込むことのできるレビュー機能があり、商品の評価や評判なども確認することができる。

また、店舗販売をメインとした企業もインターネットショッピングへの進出をしており、店舗だけではカバーしきれない品揃えを実現している。

### 2.1.1 ショッピングサイトの使い方

ここではショッピングサイトの利用方法について解説する。

#### (0. ショッピングサイトへの会員登録)

1. 商品を検索する
2. 目的の商品を見つけたら買い物かご（カートとも表記される）に入れる
3. 買い物かごから購入ページへ移動する
4. 登録した ID やパスワードを入力しログイン処理を行い、本人確認を行う
5. 届け先、商品の数量を入力、確定する
6. 支払方法を決定する
7. 確認画面で注文内容に間違いがないか確認する
8. 間違いがなかったら注文確定ボタンをクリックし注文完了

以上がショッピングサイトでの買い物の手順である。これはほしいものが明確に決まっている際の流れであるが、現在購入する予定の商品ではないが後日購入したい、またはこの商品が気になるというものは、手順 2 よりほしいものリストというお気に入り商品一覧ページに登録しておくことにより、いつでもほしいものリストから登録商品のページに移動することや、直接買い物かごに入れることが可能である。

手順 0 にて記入したショッピングサイトへの会員登録は、利用するショッピングサイトによって登録しないと買い物ができないところや、登録せずゲストとして買い物をできるところがある。手順 4 以降の流れは会員登録を行った場合の流れだが、会員登録を行わない場合手順 4 は省かれ、手順 5 は届け先の確認ではなく名前や届け先などの個人情報入力、確認となる。

### 2.2 ショッピングサイトの支払い形態

ショッピングサイトの支払い方法には、大きく分けて代金引換、コンビニ支払い、クレジットカード支払いの 3 種類が存在する。この 3 種類それぞれの特徴、メリット、デメリットについて考察する。

### 2.2.1 代金引換

代金引換支払いの特徴は、代金は商品を配達している運送業者に支払うという点である。

メリット

- ・商品が届いたときに現金で支払うため金銭の管理が容易である
- ・事前に支払いに行くというアクションがない

デメリット

- ・送料とは別に手数料が取られるサイトがある
- ・支払いのタイミングは商品が届いたときであるため家にいるときでないと支払えない

以上のように現金で支払うためお金の管理がしやすいが、家にいることが前提になってしまうため、時間指定ができないと少々不便である。

### 2.2.2 コンビニ支払い

こちらは代金引換と違い、代金をあらかじめコンビニやATMなどで直接支払うのが特徴である。

メリット

- ・コンビニやATMで手元から直接お金を支払うことができるためお金の管理がしやすい
- ・立ち寄り先のコンビニで気軽に支払える

デメリット

- ・支払いが確認されないと商品が発送されないため事情により支払いに行けない状況が出来てしまった場合発送が大幅に遅れる場合がある

こちらは代金引換と違い、コンビニに支払いに行かなければいけないため、コンビニに立ち寄る暇がない場合などには使いづらい支払い方法である。

### 2.2.3 クレジットカード支払い

こちらは直接支払うものではなく支払いはカード会社が支払いを代行し、後日登録したクレジットカードの支払い口座から引き落とされるものである。

メリット



- ・支払いに行くアクションを起こさなくても自動で引き落とされる
- ・クレジットカード情報をショッピングサイトに登録しておくため、初回以降の買い物はスムーズに行うことが出来る

#### デメリット

- ・支払った段階でいくら使ったのかは確認できるが、月別に支払うことになるのでいくら使っているのか、また支払い時の残高がいくら必要なのか気軽に確認できない
- 支払いに関しては気軽に行えるが、最も注意を払う必要がある。

### 2.3 大学生の平均生活費

私たち私立大学生の平均の生活費は、年約 62 万円で、内住居費や光熱費、食費が約 34 万円となっており、保健衛生や、娯楽嗜好、日常生活に使える額は約 28 万円となっている。(文献[9])

しかし、この中で買いものに割ける金額は約半分になり、1 ヶ月ごとになるとクレジットカードの利用上限最低金額の 5 万円にも満たない。

### 2.4 問題点

以上の支払い方法を比較してみると、安全なのは代金引換、コンビニ支払い、容易なのはクレジットカード支払いである。しかし、クレジットカードでは支払日にその月使った分を引き落とされる。そのため考えずに使いすぎて場合によっては手元にある以上の金額を使ってしまう人も存在する。加えてネットショッピングにおけるクレジットカード支払いは、店舗でのクレジットカード支払いのように毎回暗証番号入力やサインなどが必要なく、初回利用時にクレジットカード情報を登録しておくことにより、それ以降の買い物では支払方法でクレジットカード支払いを選択して購入確定ボタンをクリックするだけ買い物が完了してしまうため、買い物をしたという実感が持ちづらい。

お金を支払いに行く時間がない、もしくはあまり家にいることがないと言っても、このような場合はクレジットカードを利用するのは危険である。

## 2.5 現状のサービスや対策

現状このような金銭管理や購入物管理という点で、どのようなサービスがあるのかを調査と分析を行う。

現状のサービスでは、金銭管理は家計簿型のサービス、購入物管理はショッピングサイトなどのほしいものリストの機能が類似しているサービスである。

### ・ほしいものリスト

ほしいものリストとは、ショッピングサイトなどに備えられている機能の一つである。商品情報を検索し、気になった商品情報をほしいものリストに登録することで、その商品情報及び商品ページへのリンクを保存することができ、ほしいものリストを開くことで今までに登録した商品の情報一覧と商品価格などを確認できる。物によってはほしいものリストを複数作成でき、ほしいものの種類や優先度順にほしいものリストを分けて管理することができるものもある。

amazon.co.jp(<http://www.amazon.co.jp/>) (文献[1])

### ・ほしい物リスト(amazon.co.jpでの名称、本論文ではほしいものリストとする)

ほしいものリストを複数作成でき、種類によって管理ができる。ユーザーによってはテーマ別にほしいものリストを作成し自分のおすすめ商品を登録、公開することにより、他のユーザーへのおすすめ商品リストとして利用することもできる

また、商品を指定しないでどのようなものが欲しいというようなことをメモしてほしいものリストに加えることもできる。

試算の機能についてはカートに移す事によって選択することができる。カート内の商品の金額合計を表示することができ、「今は買わない」を選択することによってその商品の代金を省いた計算も可能。そのまま注文確定することも出来る。

### メリット

- ・複数ほしいものリストが作成できるためリストを整理できる
- ・ボタンひとつでカートに商品を入れ、購入することができる
- ・ほしいものリストの公開非公開を設定できるためおすすめ商品リストとして使うこともできる

### デメリット

- ・同じ商品の出品者が多数いる場合その商品自体をほしいものリストに追加するため

amazon が直接出品しているもの以外を買うときは商品情報のページでどこから購入するか指定しなければならない

なんでもほしいものリスト

Amazon のほしいものリストの機能の一つ、他のサイトの商品ページの商品もほしいものリストに登録できる。やはり自社サイトの商品でないため直接購入できないが、同系統商品でも値段などを比べたい時に並べられるのは便利である。

### 新しいほしい物リスト

このリストはほかの人が閲覧することはできません。公開/非公開の設定を変更する

[このリストを管理する](#)  [このリストを印刷する](#)

ページ 1 / 1 (3 商品)



アイデアを保存して、あとでお買い物しましょう。

[ほしい物リストに追加する](#)

あなたのほしい物を記録しておきましょう。  
例:  
「コーヒーマーカー」、「赤いスカーフ」、  
「旅行用のマグカップ」

---



**フラマファニチャー 低ホルムアルデヒド ダブルスライド書棚奥深タイプ ブラウン KSS-0008EBR**  
フラマファニチャー

★★★★☆  (6)

¥ 6,480  
在庫あり。出品者: Amazon.co.jp

4点の新品/中古品 ¥ 6,480より

[ショッピングカートに入れる](#) [ほかのリストに移す](#) | [削除する](#)

---



**山善(YAMAZEN) 大容量ダブルスライド本棚 CSCS-9090(BK) ブラック**  
山善

★★★★☆  (28)

¥ 5,000  
在庫あり。出品者: Amazon.co.jp

2点の新品/中古品 ¥ 5,000より

[ショッピングカートに入れる](#) [ほかのリストに移す](#) | [削除する](#)

図 2.5-1 amazon ほしい物リスト

コメント、数量、優先度を追加する 閉じる


 フラマファニチャー 低ホルムアルデヒド ダブルスライド書棚奥深タイプ  
 ブラウン KSS-0008EBR  
 フラマファニチャー

コメント (250文字まで)

欲しい数量:   
 受け取った数量:   
 優先度

図 2.5-2 ほしい物のコメント、優先度設定画面

楽天市場(<http://www.rakuten.co.jp/>) (文献[10])

- ・ 楽天お気に入りブックマーク

楽天のほしいものリスト、公開非公開の設定ができ、ボタンひとつで登録できる。登録してから持っている、持っていないというタグ設定や、チェックした商品の削除、比較などができる。こちらは基本的に自社商品が基本価格として表示されるamazon.co.jp に対して、登録された商品が大量にあり、同じ商品でも価格が違うものが多く存在するため、商品を並べて比較できるのはとても便利である。

メリット

- ・ 複数ほしいものリストが作成できる
- ・ 同一商品でも出品者別に商品情報がわかれているためどこの商品を購入したいかというところまで考えてほしいものリストに入れることができる

デメリット

- ・ 商品比較は便利だが比較するのは値段だけであり一商品ずつしかできないため同一商品で使う場合はこの機能を使用するより見比べた方が早い



図 2.5-3 楽天お気に入りブックマーク

- ・家計簿、お小遣い帳
- ・家計簿、お小遣い帳（紙媒体）

これは web サービスではないが、金銭管理システムの根幹となり得るものなので分析結果について述べる。お小遣い帳は、基本的に紙媒体で収入、支出の内訳を記入し、自分の小遣いや家計の管理を行うためのツールである。正確に記入しておけば金銭の管理には大いに役立つ。

書店などで購入することができるが、マイクロソフト社の Excel（以下、エクセル）などでフォーマットを自作し、印刷して使用することもできる。クレジットカードの利用金額もこちらで記入し、引き落とされる分の金額を来月の内訳に記入し管理することで大きな間違いは起こりづらくなる。

また、種類によっては貯金目標金額の設定や、ほしいものを目標とするもの、クレジットカードなどの管理をしやすい形で作られているものもある。

#### メリット

- ・機械などの苦手な人でも難なく利用できる

#### デメリット

- ・持ち歩くことがあまりないのでその場で気軽に入力ということができない
- ・正確に計算するためにはレシート等の正確な記録が必要になる
- ・計算を間違えると大きな誤差が生じる

また、PCでの利用も前述でのエクセルなどである程度の計算機能を組み込んだ家計簿の型紙データを作成し、ある程度の集計を自動化させた家計簿として利用することもできる。

- ・いいめも (<http://www.ememo.jp/okodukai/>) (文献[8])

携帯電話を使った簡易お小遣い帳サービス。

使い方はメール登録してから届いたアドレスに買ったものの名前と金額を入力してメールを送る。そうすると何にいくら使ったかということをサイトで記録してくれるサービスとなっている。

累計金額のリセットはできないが、月はじめから改めて数えなおしてくれるので月別の使用金額は算出できる。

#### メリット

- ・シンプルに使用した金額が入力できる
- ・メールに使用した金額と商品名を打って送信するだけなので気軽に登録できる。

#### デメリット

- ・手持ちを入力して残額計算、試算などはできない。
- ・登録情報が個別にリストとして表示されるので内訳を詳細に入力しているとリストが長く見にくいものになってしまう。



図 2.5-4 送信画面

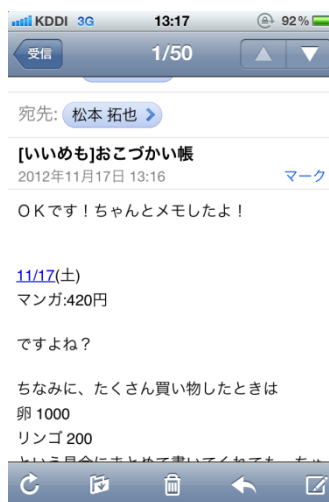


図 2.5-5 受信画面

このように「買ったもの 値段」という形で「いいめも」にメールを送信するとすぐに登録結果のメールが帰ってくる  
 これを取り消したい場合は「とりけし」とだけ本文に入力し「いいめも」に送信すると

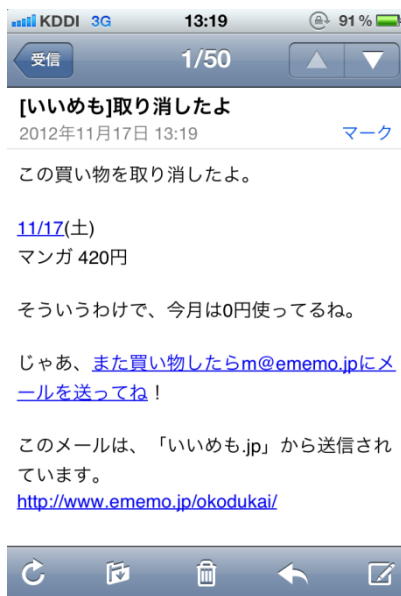


図 2.5-7 取り消し操作を行った時の受信メール

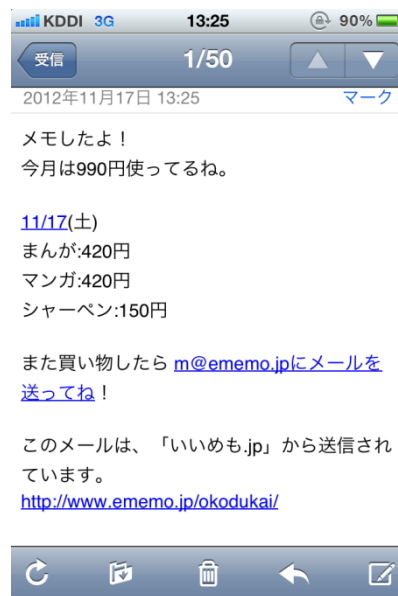


図 2.5-6 複数回登録を行った際の受信メール

図 2.5-7 のメールが返信され取り消しが完了する。

複数情報を登録したい時は「買ったもの1 値段  
買ったもの2 値段」

という形で改行を入れ送信をすることで複数の登録を一度にすることができる。

合計金額は2回目以降の情報登録から表示されるため一回目では複数情報を登録しても合計金額は表示されない。複数回情報を登録した場合、図 2.5-6 のように表示される。

しかし、「いいめも」は買ったものを登録した時に今までの買ったものリスト、値段、月に使った合計金額を表示してくれるが、月の利用金額の確認だけの目的で利用することはできない。

#### ・お財布サイト

WebMoney (<http://www.webmoney.jp/>) (文献[7])

ウェブマネーというインターネット上の取引で利用できるプリペイドサービスサイト。

ウェブマネーとはインターネット上で使える通貨で、基本的にプリペイドサービスの形をとっている。一部のショッピングサイトやオンラインゲームの課金、またダウンロードソフトの購入などが主な利用先である。

利用の仕方はコンビニなどでウェブマネーを購入し、このサイトにチャージすることにより残額を追加することができる。WebMoney の単位は Point と表され、基本的に日本円と同価値である。

また、ウェブマネーを利用した支払いは、通販サイトなどの支払い方法でウェブマネーを選択し、ウォレットにログインすれば支払いは完了する。マイページには最近の利用明細として、チャージした金額や利用履歴などが表示される。

また、メール利用明細通知や CSV 形式での利用履歴のダウンロードも出来る。

#### チャージ方法

##### ・プリペイド支払い

コンビニや取り扱い店舗でウェブマネーカードを購入してカードに表記されているプロダクトコードを入力することでチャージを行う。基本的にレジでの先払い。



- ・インターネットバンキング  
ネットバンクの残高をウェブマネーに変換してチャージする方法
- ・クレジットカード支払い  
クレジットカードを利用してチャージを行う
- ・セブンイレブンでのインターネット収納代行

#### メリット

- ・プリペイド支払いでの利用をすれば予めウォレットにお金を移しているため口座の金額が足りないということが起こらない。
- ・誰でも利用することができる。

#### デメリット

- ・予めプリペイドポイントを購入する必要がある
- ・買い物をした際中途半端な金額のものを買ってしまうと残高ポイントを使い切るのが難しくなる
- ・利用できるサイトが少ない（ただしオンラインゲームなどでは主流の支払い方のため、そちらで利用する分には特に苦勞はない）

#### Vプリカ(<http://vpc.lifecard.co.jp/>) (文献[6])

VISA を利用したプリペイドサービスで、これは支払いなどがクレジットカードを利用した支払いと同じような使い方のできるプリペイドサービスと考えていい。

利用方法はコンビニやクレジットカードを利用して V プリカのプリペイドマネーを購入してサイトでチャージを行い、発行されたマネーを使って買い物を行う。利用する際にはクレジットカード利用の VISA を選択し、クレジットカード番号を入力する欄にこのサイトで発行されたクレジット番号を入力することにより利用できる。以後買い物をする際にはクレジットカード支払いを選択するだけでこちらのサイトから引き落とされるようになる。

#### メリット

- ・簡単にアカウント作成できる（審査などが必要ない）

- ・プリペイド形式となっているので、買い物をしたら即 V プリカの残高から引き落とされるため、支払いのタイムラグが生じない
- ・ネットでの支払いではクレジットカードと相違なく利用できる

#### デメリット

- ・カードは発行されないため、ネットショップ以外（店舗での買い物など）では利用できない、あくまでインターネット上でのものである。

表 2.5-1 にてこれらのサービスをクレジットカードと比較すると、

	審査	支払い形式	利用範囲	認証方法
クレジットカード	あり	口座引落	登録店舗、ネット	カード番号等※
WebMoney	なし	プリペイド	一部ネットショップ	ID 等、暗証番号
V プリカ	なし	プリペイド	ネットショップなど	カード番号等※

表 2.5-1 クレジットカードとお財布サービスの比較

※カード番号と暗証番号、ネットショッピングや一部のサービスのみクレジットカードと V プリカは同様の操作、扱いで利用できる

お財布サービスは利用できる場面が限られているが、ネットショッピングなどで気軽に利用することができる。また、最大の違いは支払いがプリペイド方式であることと、支払いにタイムラグが発生しないことである。

お財布サイトに分類したサービスは、クレジットカードを持っていない人にとっては便利ではあるが、外での買い物に利用できないことを考えるとやはりクレジットカードの補助的役割でしかない。

ネットショッピングでの金銭管理をするにあたり、ほしいものリストとお小遣い帳、家計簿を合わせたようなものが必要である。

### 3. 解決法の検討

#### 3.1 解決すべき問題

ここでは、特にクレジットカードでの支払い、その中でも月ごとの支払い額の管理のしにくさに焦点を当て、問題点を明確にする。

まず、支払い額の管理がしにくい理由として

- ・利用額確認を行わなければいけない
- ・購入直後にカード会社のサイトで利用額確認しようとしても反映に時間がかかる
- ・買い物をしても即座に引き落とされないため、月別の支払額を管理しないと大幅なずれが生じる

これらを改善する方法を以下の方法により解決のアプローチの検討を行う

#### 3.2 解決へのアプローチ

上記で述べた各問題の解決策を提案・検討する

##### ・支払額確認の問題

口座残額確認などは、収入があった時などに確認するため、その時に通帳への記帳などという形で記録をしておくことが望ましい。また、銀行によっては web で残額や利用履歴の確認、出力、メールで明細書を送る場合があるため、そちらを利用するのもいい。

##### ・金銭管理の問題

利用額計算は購入したい商品の値段などの情報を、API を利用し抽出し、それに加え手数料などを含めた額を加え計算するプログラムを作成する。予算又は上限額に関しては手入力により入力してそこから減算を行い、残額を計算する。また、予算残額は月ごとにリセットを行い、次の買い物の際、新たに予算残額からの減算を行うのが望ましい、管理を目的とするため残額を計算するだけでなく、予算、利用限度額を下回った場合（予算残額から購入金額を減算していった結果その金額が 0 を下回った場合）、警告文を表示する。利用限度額などはデータベースなどで管理する。

## 4. システムの設計と実装

この章ではシステムの設計と実装を行う。

### 4.1 システム設計

解決へのアプローチをもとに、具体的にどのような機能が必要かをここで述べる。

- ・ API を利用した商品検索機能及びほしいもののリストの作成

ショッピングサイトの API を利用した商品検索機能で商品情報を取得し表示、その中で気になるものをほしいもののリストへ移すことによって購入予定、検討中の商品を管理しやすくする。

- ・ 試算機能における予算及びカード利用上限額の計算機能

はじめに計算を行うとき、残高及びカード利用上限額の入力を行うことで、今月はあといくら使えるのか、また今月の利用額はいくらなのかを管理しておく必要がある。はじめにこちらに上限額の入力を行ったらあとはシステムのほうで残額を管理し、上限額に達するまであといくらか、また今月の利用額を明記することにより管理をしやすくする。

- ・ 予算及びカード利用上限額の減算、リセット

クレジットカードが利用できるのはインターネット上だけでなく、日常の買い物でも利用ができるため、手動入力で残高の減算をできるようにすることにより、より正確に利用額を管理できるようにする。また、利用額を減らしたことに對してほしいもののリストに購入済み扱いで手動入力という内訳を作成することで何に使ったかを明確に表示する。

また、利用上限額は月ごとにリセットされるため、こちらも月ごとに入力をし、修正できるようにする。リセット操作を行うことにより月の利用額を締め、先月の利用額という管理に移し、新たに利用額管理を行う。これにより、予算及び利用上限額をリセットするごとに月が切り替わることにより、より計画的な買い物を支援する。

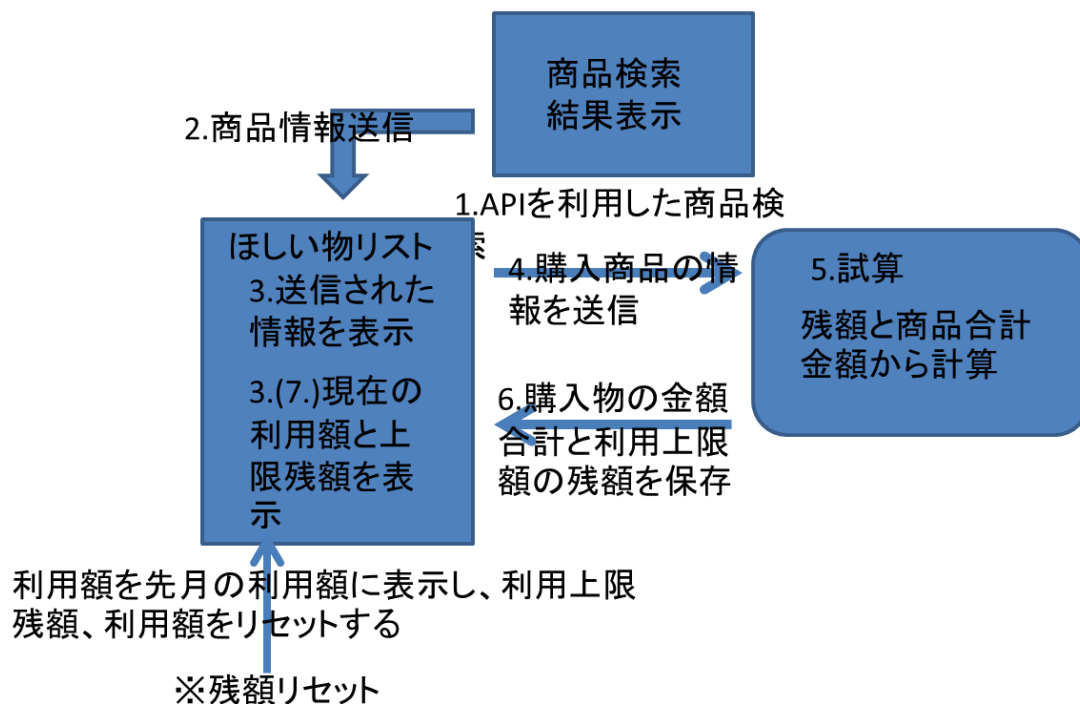


図 4.1-1 システムの動き

## 4.2 データベースの設計

このシステムを実現するにあたって、必要になった複数のデータベースを設計する。

### 4.2.1 アイテムコード保存用データベース「itemcode」

このデータベースは、商品検索機能で取得した商品情報の中から選択した商品のアイテムコードを保存するために利用する。ほしいもののリストでは連番を主キーとしてアイテムコードの参照を行うため、連番とアイテムコードを保存することが必要である。

テーブルを作成するにあたり必要なテーブルは、前述の連番とアイテムコードを保存するものである。連番フィールドは名前を「renban」とし、属性は数値を保存する int 型を使用する。また、連番として通し番号を自動で振るようにする。次にアイテムコードを保存するフィールドは名前を「itecode」とし、こちらは文字として保存するため char 型を利用する。今回使用する楽天市場でのアイテムコードは長いものが存在するため、char(50)を使用する。

#### 4.2.2 計算用数値保存用データベース

計算用の数値を保存するデータベースを 3 つ作成する。これらはそれぞれ予算、残額、合計額、先月の利用額を保存するため利用する。これらすべてのデータベースの目的は、それぞれの用途に応じた数値を、最新レコード 1 つだけを保存しておくために利用するため、連番などは利用せず数値を保存するためのフィールドだけを設ける。

フィールドの型は数値の保存を目的とするため int 型を利用する。テーブル名、フィールド名、はそれぞれ

- 予算データベース

テーブル名 : yosan

レコード名 : yosan (int 型)

- 残額データベース

テーブル名 : zagaku

レコード名 : zagaku (int 型)

- 先月の利用額データベース

テーブル名 : riyougaku

レコード名 : riyougaku(int 型)

とする。

#### 4.3 システムの実装

このシステムではほしいものリストを実装するにあたり、検索システムを構築し、その検索システムにより抽出したアイテムコードをデータベースに保存し、保存されたアイテムコードから商品データを検索、取得したものをほしいものリストとして表示するため、アイテムコードを保存するデータベースを作成した。

設計したテーブルの構築画面

```
mysql> use db_a940186;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table itemcode(renban int,itemcode char(50));
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> alter table itemcode add index ind1(renban);
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> alter table itemcode modify renban int auto_increment;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

このコードにより、データベース「itemcode」を作成し、renban と itecode というフィールドを作成した、また、renban には主キーの役割を与えるため自動で通し番号が当てられるようにした。

図 4.3-1 itemcode のフィールド

Field	Type	Null	Key	Default	Extra
renban	int(11)	NO	MUL	NULL	auto_increment
itecode	char(50)	YES		NULL	

#### 4.3.1 検索機能

(<http://www.cyaneum.org/~a940186/sotsuken/sotsuken-rakuten2.php>)

この検索機能では、楽天市場の API を利用し商品を検索し、取得した商品情報を表示する。表示した商品のなかでほしいものリストに追加したいものがあつたら「ほしいものリストに追加」ボタンを押すことで商品登録ページに飛び、データベース「itemcode」への保存を行う。

検索キーワード:

検索する

Supported by [楽天ウェブサービス](#)

図 4.3.1-1 検索

検索キーワード:

検索する



製品名:【レビューを書いてお買い得!】【メール便不可】DARTS HIVEダーツハイブ 『L-Caseが付いたお洒落なダーツケース』【メール便不可】CAMEO COLORS with L-caseダーツ ソフトダーツ ダーツケース コンパクトケース カメオ カラーズ L-style フライト・エル エルフライト・エルケース【Darts shop Hive(ダーツ ハイブ)】(ダーツケース/case/楽天)

商品コード: hive:10008817

価 格: 3400

店 名: DARTS SHOP Hive (ダーツ ハイブ)

ほしいものリストに商品を追加



製品名:【レビューを書いてお買い得!】【メール便OK】DARTS HiVeダーツハイブ コンパクトなダーツケース新登場!【メール便OK】SLOW HAND バレルホルダーソフトダーツ ダーツ Barrel holder 首掛けケース 腰掛け アクセサリー ダーツケース コンパクト カバンに収納 090817sh【darts shop Hive(ダーツ ハイブ)】(ダーツケース/case/楽天)

商品コード: hive:10008712

価 格: 1470

店 名: DARTS SHOP Hive (ダーツ ハイブ)

ほしいものリストに商品を追加

図 4.3.1-2 検索結果



## プログラム

このプログラムは、テキストボックスに入力されたキーワードを POST 形式で受け取り、キーワードを利用して楽天 API を利用し検索を行う。そして楽天市場の検索により取得された商品情報の内、5 つの商品の画像、商品名、商品コード、価格、販売店名を表示し検索結果として表示する。また、各商品情報の下に「ほしいものリストに商品を追加」というボタンを設置し、それをクリックすることで、その商品のアイテムコードを `touroku.php` に GET 形式で送信する。

### sotsuken-rakuten2.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Rakuten API Shopping To Wish</title>
  </head>
  <body>
    <?php

//楽天から商品を検索して HTML タグを返す関数
function search_rakuten($keyword,$limit){

  //developerId を設定する
  $developerId = '*****';

  //affiliateId を設定する
  //$affiliateId = '取得したアフィリエイト ID'; //なくてもよい

  //使用する API の操作名を設定する
  $operation = 'ItemSearch';
```

```
//バージョンを設定する
$version = '2009-04-15';

//取得件数を設定する
$hits = $limit;

//検索キーワードを URL エンコードして設定する
$text = urlencode($keyword);

//人気の高い順に検索する
$sort = urlencode("standard");

//URL を生成する
$url = 'http://api.rakuten.co.jp/rws/3.0/rest?'.
    'developerId='.$developerId.
    '&affiliateId='.$affiliateId.
    '&operation='.$operation.
    '&version='.$version.
    '&keyword='.$text.
    '&hits='.$hits.
    '&sort='.$sort;

//REST で返される XML ファイルを取得する
$str = file_get_contents($url);

//XML を解析してオブジェクトにセットする
$xml = simplexml_load_string($str);

//xpath によって、Item 要素だけを取り出す
```

```

$Items = $xml->xpath('//Items/Item');

//div タグでくくる
$ret = '<div id="rakuten_item">';

//Item 数だけ繰り返す
foreach($Items as $Item){
    $ret .= '<a href="'. $Item->affiliateUrl. "'>';
    $ret .= 'itemName. "'>';
    $ret .= "</a>¥n";
    $ret .= "<p>製品名 : " . htmlspecialchars($Item->itemName) . " <br />";
    $ret .= "<p>商品コード : " . htmlspecialchars($Item->itemCode
    ) . " <br />";
    $ret .= "価 格 : " . htmlspecialchars($Item->itemPrice) . " <br />";
    $ret .= "店 名 : " . htmlspecialchars($Item->shopName) . " </p>";
    //ほしいものリストへ送るためのチェックボックスを作成
    $ret .= "<form action='touroku.php' method='GET'>
    <input type='hidden' name='ItemCode' value = $Item->itemCode >
    <input type='submit' value='ほしいものリストに商品を追加'/>
    </form><br>";
}
//div タグを閉じる
$ret .= '</div>';

//作成した HTML を返す
return $ret;
}

```

```

echo "
<p>
<form method=¥"POST¥">
<p>検索キーワード :
<input type=¥"text¥" name=¥"keyword¥" value=¥"¥" />
</p>
<input type=¥"submit¥" value=¥"検索する¥" />
</form>
";

//form から post で送られたデータを受信する。

extract($_POST);

if($keyword<>"){
    //取得数を設定する
    $limit = 5;

    //楽天から商品を検索して表示する
    echo search_rakuten($keyword,$limit);
}
?>
<a href = "rist.php">ほしいものリストへ</a>
<br>
<!-- Rakuten Web Services Attribution Snippet FROM HERE -->
<a href="http://webservice.rakuten.co.jp/" target="_blank">Supported by 楽天ウェブサービス</a>
<!-- Rakuten Web Services Attribution Snippet TO HERE -->
</body>
</html>

```

### 4.3.2 登録

検索結果の「ほしいものリストに商品を追加ボタン」を押すことにより、商品のアイテムコードを GET 形式で登録ページへ送り、こちらのページで SQL を利用しアイテムコードを登録する。また、ページに設置してあるハイパーリンクを利用することでほしいものリストに移動や検索ページに戻ることができる。

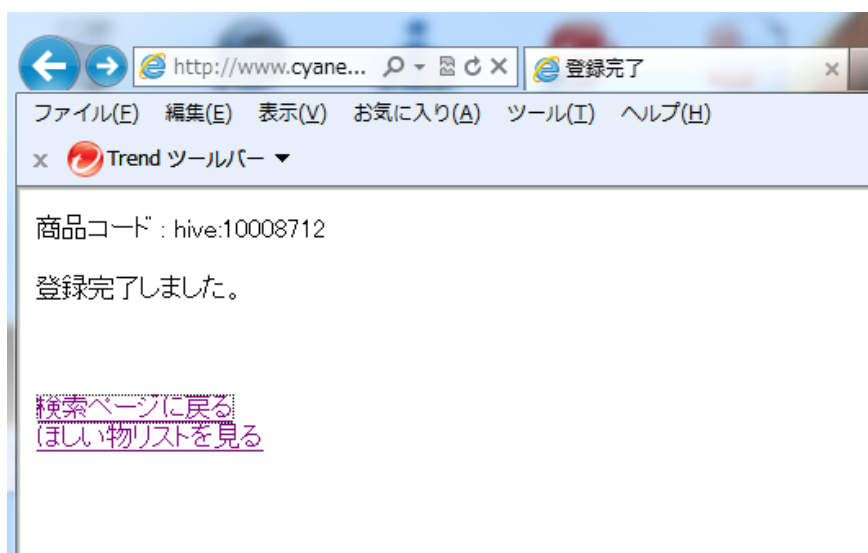


図 4.3.2-1 登録ページ

renban	itecode
1	hive:10008712

図 4.3.2-2 データベース登録結果

プログラム

sotsuken-rakuten2.php から送られてきたアイテムコード情報を受信し、画面内に表示することにより、どんなアイテムコードをデータベース「itemcode」に登録したのか確認できるようにし、そのアイテムコードの保存されている変数を利用し、mysql を起動した後アイテムコードをデータベース「itemcode」に保存するという指示を送り、データベース「itemcode」にアイテムコード情報を登録する。

touroku.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
    <title>登録完了</title>
</head>

<body>

<?php
    //登録するアイテムコードを表示する、itemcode の数だけ繰り返す
    foreach($_GET as $index => $itemCode){
        //if ($_GET as $index => $itemCode) ++$_SESSION['itemCode'];
        echo " 商品コード : $itemCode<br>";
    }

    //SQL を利用しデータベースにアイテムコードを登録する
    mysql_connect('localhost','a940186','*****');
    mysql_select_db('db_a940186');
    $sql = "insert into itemcode values(0,'$itemCode)";
    mysql_query($sql);
?>
<p>登録完了しました。</p><br>
<br>
<a href = "sotsuken-rakuten2.php">検索ページに戻る</a>
<br>
<a href = "rist.php">ほしいものリストを見る</a>

</body>
```

</html>

#### 4.3.3 ほしいものリスト、試算機能([www.cyaneum.org/~a940186/sotsuken/rist.php](http://www.cyaneum.org/~a940186/sotsuken/rist.php))

ほしいものリストと試算機能は、基本の予算金額を表示する「yosan」、残額を登録する「zangaku」、先月の表示額を登録する「riyougaku」の3つのデータベースを利用して予算、残額、先月の利用額などに分けて管理、計算をする。

仕組みは検索ページから登録された「itemcode」のデータベースのレコードを参照し、楽天APIによりアイテムコード検索をかけ、検索結果をほしいものリストとして表示する。

商品情報の削除は、削除用のページにて削除したい商品情報を選択し、削除実行のボタンをクリックすることにより、指定された商品のレコードをデータベース「itemcode」から削除することができる。

ほしいものリスト下部に表示されている残額表示用テキストボックスには、データベース「yosan」を参照し登録されたデータベース「zangaku」より参照した数値を表示する。また、計算を行うたびに購入物の金額分減算していき、「zangaku」に登録されるため、購入計算を行うたびに更新され表示される。

リセットを行うと「zangaku」に入力されている数値を「yosan」から減算し、計算結果を「riyougaku」に登録する。「zangaku」のレコードを削除した後、「yosan」の数値を「zangaku」に登録する。また、この際リセットボタンの横のテキストボックスに別の数値を入力することにより、「yosan」データベースの数値を変更することができる。

計算は購入したい商品をチェックボックスにより選択し、「購入シミュレーションをする」ボタンをクリックすることにより、残高のテキストボックスに入力されている数値と選択した商品の合計金額を用い残額の残りを計算する。計算結果表示後、残額確定のハイパーリンクを押すことにより、計算に利用した商品を購入したとみなし、「zangaku」に計算後の数値を反映する。反映しない場合は、残額確定をクリックせずに更新や検索ページなどからほしいものリストに戻るなどの操作を行うことで元の表示へ戻ることができる。

### 計算のイメージ

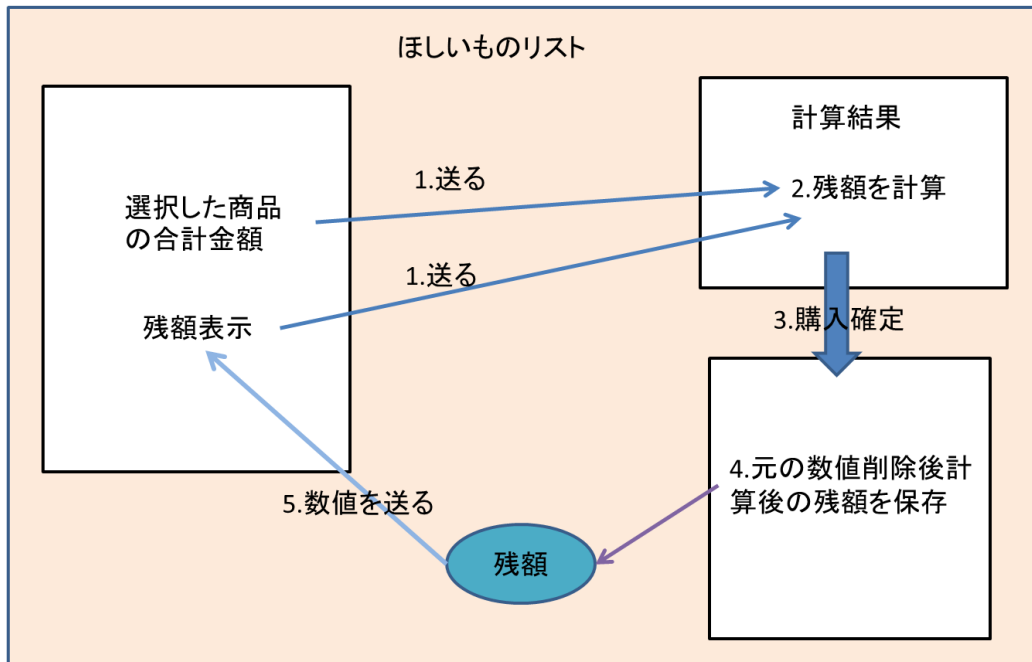


図 4.3.3-1 計算のイメージ

### リセットのイメージ

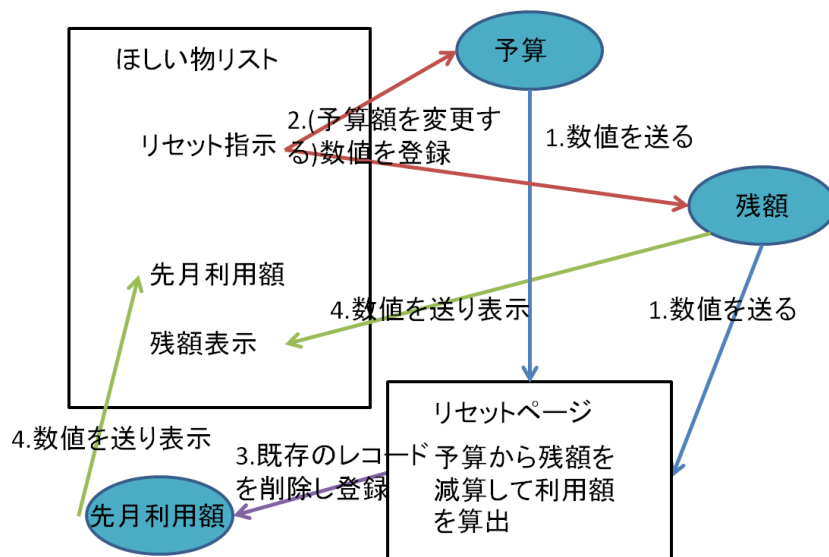


図 4.3.3-2 リセットのイメージ



この機能のために作成した 3 つのデータベースは、最新のレコードだけを保存しておくものとし、保存するものは先月の利用額、残額、予算のそれぞれに対応した数値だけである。

「riyougaku」, 「zangaku」, 「yosan」 テーブルの構築画面

```
mysql> use db_a940186;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table riyougaku(riyougaku int);
Query OK, 0 rows affected (0.00 sec)

Database changed
mysql> create table zangaku(zangaku int);
Query OK, 0 rows affected (0.00 sec)

Database changed
mysql> create table yosan(yosan int);
Query OK, 0 rows affected (0.00 sec)
```

Field	Type	Null	Key	Default	Extra
riyougaku	int(11)	YES		NULL	

図 4.3.3-3 riyougaku のフィールド

Field	Type	Null	Key	Default	Extra
zangaku	int(11)	YES		NULL	

図 4.3.3-4 zangaku のフィールド

Field	Type	Null	Key	Default	Extra
yosan	int(11)	YES		NULL	

図 4.3.3-4 yosan のフィールド

#### 4.3.3.1 ほしいものリストと計算機能

検索機能と登録機能により登録したアイテムコードをデータベース「itemcode」から受け取り、楽天 API より検索し、検索結果をほしい物リストとして表示をする。また、ページの下部にある「商品情報を削除」のハイパーリンクをクリックすることにより、商品削除ページへ移動することができる。

他にも、先月の利用金額や残額、予算などを各データベースより受け取り、テキストボックス内やテキストとして表示する。テキストボックスに表示されているものは、変更することもできるという事を前提に作っており、残額ならば店舗でクレジットカードを利用した際の利用額分の減算、予算のリセットは来月の予算、利用上限の増減に対応するためのものである。何もなければそのまま「商品購入シミュレーションをする」ボタンや「上限額をリセットする」ボタンを押せばそれぞれの動作を実行する。

また、残額の計算もこのページでできるようになっており、計算に利用したい商品のチェックボックスにチェックを付け、「計算シミュレーションをする」ボタンをクリックするとこのページに商品の合計金額と現在の残額の 2 つの数値を計算し、結果を表示する。

計算の結果、残額が 0 を下回らなければ残りの金額と「計算結果を反映」のハイパ

ーリンクが表示されるが、残額が 0 を下回った場合には計算結果は表示されず赤文字の警告文が表示される。

計算結果を反映するには計算後警告が表示されていない場合に限り、計算結果を反映するためのハイパーリンクが表示される。これをクリックすることで、残額更新のページへ移動する。

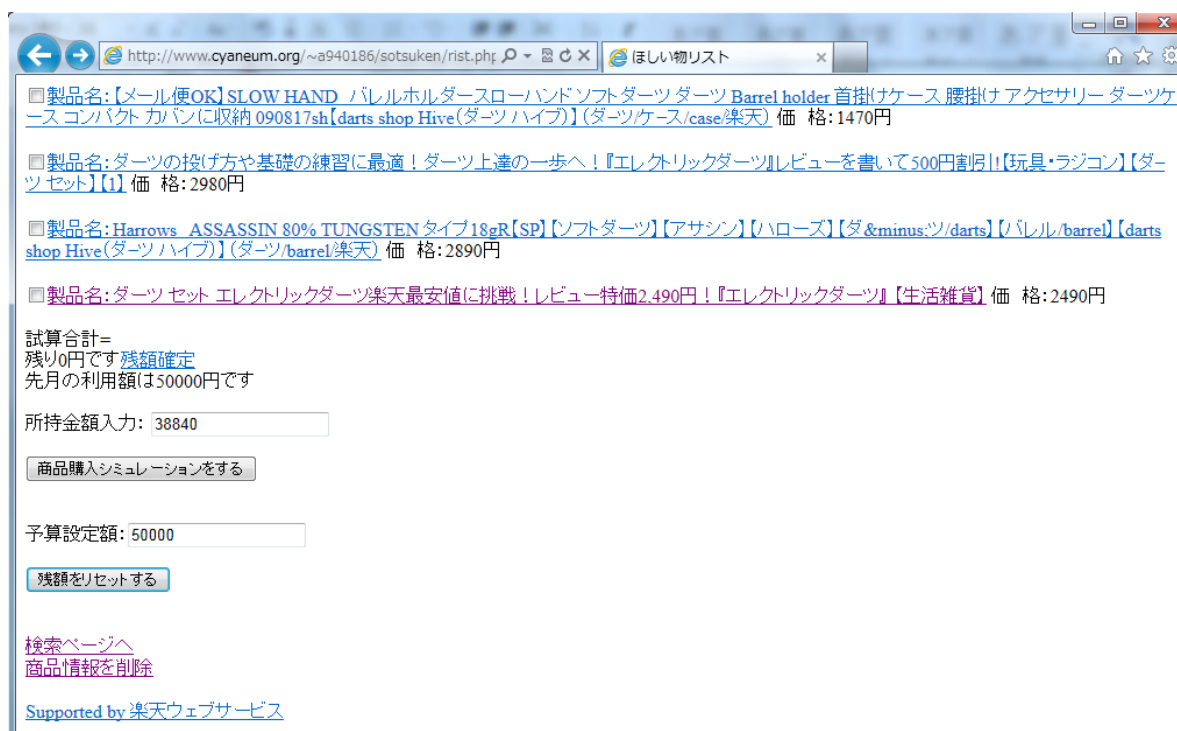


図 4.3.3.1-1 ほしい物リストの動作画面

図 4.3.3.1-1 では予算額に関して、実際にクレジットカードを使用するにあたり、警告ラインが低すぎると買い物がしづらくなるため、暫定的にクレジットカードの最低利用額である 50000 円を予算と設定してある。

商品情報の左に設置してあるチェックボックスをチェック（下の商品 2 つにチェックを入れる）して「商品購入シミュレーションをする」ボタンを押すと図 4.3.3.1-2 のような結果が表示される。ここで「残額確定」のハイパーリンクをクリックすることで、計算後の残額をデータベースに反映するためのページへ移動する。また、残額が 0 円を下回ってしまった場合には図 4.3.3.1-3 のような結果が表示される。

試算合計=5380  
残り33460円です [残額確定](#)  
先月の利用額は50000円です

所持金額入力:

[商品購入シミュレーションをする](#)

図 4.3.3.1-2 ほしいものリスト拡大・計算結果

試算合計=2490  
残り -1490 円です  
使用限界ラインを超えています！！購入するか再検討してください。  
先月の利用額は50000円です

所持金額入力:

[商品購入シミュレーションをする](#)

図 4.3.3.1-3 ほしいものリスト拡大・計算結果、残額が 0 を下回った場合  
プログラム

データベース「itemcode」からアイテムコード情報を取出し、取り出したアイテムコードを楽天 API のアイテムコード検索にかけ、受け取った結果をほしいものリストとして表示する。ほしいものリストとして商品情報の商品名、価格のほかに、計算対象に選択するためのチェックボックスを設置し表示する。また、画面内のテキストに先月の利用額を表示するが、データベース「riyougaku」からレコードを取出し、\$riyougaku に代入し表示する。

現在の残額や予算をテキストボックスに表示する仕組みもそれぞれデータベース「zangaku」と「yosan」からレコードを取出し、\$zangaku と \$yosan に代入を行いテキストボックスに表示する（テキストボックスに表示される数値は手動で変更して利用することもできる）。

計算機能はチェックを付けた商品の金額を\$sum に加算していき「合計金額=〇〇」

という形で表示し、合計金額を\$zangaku から減算し、計算結果を\$zangaku に代入する。その後\$zangaku が 0 以上なら「残り〇〇円です」の表示と残額確定のハイパーリンクが表示される。\$zangaku が 0 以下なら赤い文字で警告文が表示されるようになっている。

また、予算設定額のテキストボックスの下に表示されている「残額をリセット」ボタンをクリックすると POST 形式で\$yosan（予算設定額テキストボックスの中身）を reset.php 送信し移動する。

rist.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>ほしい物リスト</title>
  </head>
  <body>
<?php

//楽天から商品を検索して HTML タグを返す関数
function search_rakuten($itemcode){

  //echo "itemcode= " . $itemcode . "<br>";
  //developerId を設定する
  $developerId = '*****';

  //affiliateId を設定する
  //$affiliateId = '取得したアフィリエイト ID'; //なくてもよい

  //使用する API の操作名を設定する
  $operation = 'ItemCodeSearch';
```

```
//バージョンを設定する
$version = '2010-08-05';

//検索キーワードを URL エンコードして設定する
//$text = urlencode($itemcode);

//URL を生成する
$url = 'http://api.rakuten.co.jp/rws/3.0/rest?'.
    'developerId='.$developerId.
    '&affiliateId='.$affiliateId.
    '&operation='.$operation.
    '&version='.$version.
    '&itemCode='.$itemcode;

//echo $url;
//exit;

//REST で返される XML ファイルを取得する
$str = file_get_contents($url);

//echo $str;
//exit;

$str = str_replace('itemCodeSearch:ItemCodeSearch',
'itemcodesearch', $str);
```

```

//echo $str;

//XML を解析してオブジェクトにセットする
$xml = simplexml_load_string($str);

//var_dump($xml);
//exit;

//xpath によって、Item 要素だけを取り出す
//$Items = $xml->xpath('//Body/itemcodesearch/Item');
$Items = $xml->xpath('//Body/itemcodesearch/Item');
//div タグでくくる
//$ret = '<div id="rakuten_item">';

//Item 数だけ繰り返す
foreach($Items as $Item){
    $ret .= "<a href=¥". $Item->itemUrl. "¥">";
    $ret .= "製品名 : ". htmlspecialchars($Item->itemName) . "</a> ";
    $ret .= "価 格 : ". htmlspecialchars($Item->itemPrice). "円<br/>";
    //$ret .= "店 名 : ". htmlspecialchars($Item->shopName). " </p>";
"<br/>";
    //$zan = $zan-$Item->itemPrice;
}

//div タグを閉じる
//$ret .= '</div>';

```

```

//作成した HTML を返す
return $ret;
}

//楽天から商品を itemcode で検索して価格を返す関数
function search_rakuten_price($itemcode){

    $developerId = '0bfde87d766966d83f2e772cf77bf68d';

    //使用する API の操作名を設定する
    $operation = 'ItemCodeSearch';

    //バージョンを設定する
    $version = '2010-08-05';

    //URL を生成する
    $url = 'http://api.rakuten.co.jp/rws/3.0/rest?'.
        'developerId='.$developerId.
        '&affiliateId='.$affiliateId.
        '&operation='.$operation.
        '&version='.$version.
        '&itemCode='.$itemcode;

    //REST で返される XML ファイルを取得する
    $str = file_get_contents($url);

    $str = str_replace('itemCodeSearch:ItemCodeSearch',
        'itemcodesearch', $str);

```



```

//XML を解析してオブジェクトにセットする
$xml = simplexml_load_string($str);

//xpath によって、Item 要素だけを取り出す
//$Items = $xml->xpath('//Body/itemcodesearch/Item');
$Items = $xml->xpath('//Body/itemcodesearch/Item');
//div タグでくくる
//$ret = '<div id="rakuten_item">';

//Item 数だけ繰り返す(itemcode に対応する商品は 1 個だけ)
foreach($Items as $Item){
    $price= $Item->itemPrice;
}

return $price;
}
extract($_POST);

mysql_connect('localhost','a940186','*****');
mysql_select_db('db_a940186');
$sql = "select * from itemcode;";
$result = mysql_query($sql);
$rows = mysql_num_rows($result);
if($rows == 0){
    echo "<p>該当データがありません。 </p>";
}
else {
    echo "<form method='post'>";
    while($row = mysql_fetch_array($result)){
        "<p>";
    }
}
}

```

```

        echo "<input type='checkbox' name='ItemCode[]' value='".
$row['itecode'] . "'>". search_rakuten($row['itecode']) . "<br>";
    }
}

//チェックした商品と入力した金額で計算を行う

$num = count($ItemCode);
if ($num > 0){
    for ($i=0; $i<$num; $i++){
        $price = search_rakuten_price($ItemCode[$i]);
        $sum += $price;
    }
}

echo "試算合計=" . $sum . "<br>";

$zangaku -= $sum;

if($zangaku >= 0){
    echo "残り". $zangaku ."円です";
    echo "<a href='koushin.php?zangaku=" . $zangaku . "'>残額確定</a><br>";
}else{
    echo "<FONT COLOR=¥\"RED¥\">残り. $zangaku .円です</FONT><br/>";
    echo "<FONT COLOR=¥\"RED¥\">使用限界ラインを超えています！！購入
するか再検討してください。</FONT><br>";
}

$sql = "select riyougaku from riyougaku;";

```

```

$result = mysql_query($sql);
$rows = mysql_num_rows($result);
if($rows == 0){
    echo "<p>該当データがありません。 </p>";
}
else {
    while($row = mysql_fetch_array($result)){
        $riyougaku = $row['riyougaku'];
    }
}
echo "先月の利用額は". $riyougaku ."円です";

if ($zangaku == 0){
    $sql = "select zangaku from zangaku;";
    $result = mysql_query($sql);
    $rows = mysql_num_rows($result);
    if($rows == 0){
        echo "<p>該当データがありません。 </p>";
    }
    else {
        while($row = mysql_fetch_array($result)){
            $zangaku = $row['zangaku'];
        }
    }
}

echo "<p>所持金額入力 :
<input type='text' name='zangaku' value=" . $zangaku . ">
</p>";

```

```

echo "<input type='submit' value='商品購入シミュレーションをする'
/></form><br>";

    $sql = "select yosan from yosan;";
    $result = mysql_query($sql);
    $rows = mysql_num_rows($result);
    if($rows == 0){
        echo "<p>該当データがありません。 </p>";
    }
    else {
        while($row = mysql_fetch_array($result)){
            $yosan = $row['yosan'];
        }
    }

echo "<form action = reset.php method=¥'POST¥'>";
echo "<p>予算設定額 : <input type='text' name='yosan' value=" . $yosan . " ></p>";
echo "<input type='submit' value='残額をリセットする' /></form><br>";
?>
<br>
<a href = "sotsuken-rakuten2.php">検索ページへ</a>
<br>
<a href = "sakujo2.php">商品情報を削除</a>
<br>
<br>

<!-- Rakuten Web Services Attribution Snippet FROM HERE -->
<a href="http://webservice.rakuten.co.jp/" target="_blank">Supported by 楽天ウェブサービス</a>
<!-- Rakuten Web Services Attribution Snippet TO HERE -->
</body>

```

```
</html>
</body>
</html>
```

#### 4.3.3.2 計算結果反映

rist.php から受け取った計算結果計算結果の数値を登録するためのページ。画面上には計算結果の反映完了を示すメッセージと欲しいものリストへ戻るためのハイパーリンクが表示される。

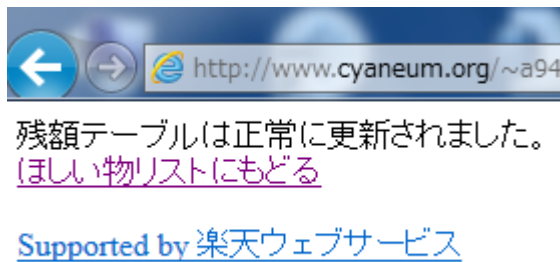


図 4.3.3.2-1 計算結果反映 動作  
プログラム

SQL コマンドを利用してデータベース「zangaku」に登録されているレコードを削除し、新たに GET で受け取った\$zangaku をデータベース「zangaku」に登録し、「残額テーブルは正常に更新されました。」というメッセージを表示する。もし該当するデータが見つからなかった場合はエラーメッセージを表示する。また、欲しいものリストへ戻るためのハイパーリンクを設置する。

kousin.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>残額の更新</title>
  </head>
  <body>
```

```

<?php

extract($_GET);

mysql_connect('localhost','a940186','*****');
mysql_select_db('db_a940186');
$sql = "delete from zangaku;";
$result = mysql_query($sql);

$sql = "insert into zangaku (zangaku) values(" . $zangaku . ")";
$result = mysql_query($sql);

if (!$result){
    $message = '無効なクエリです。: ' . mysql_error() . "<br>";
    $message .= 'クエリの内容: ' . $sql;
    echo "<p>";
    echo $message;
    echo "</p>";
} else {
    echo "残額テーブルは正常に更新されました。 <br>";
}

?>

<a href = "rist.php">ほしい物リストにもどる</a>
<br>
<br>
<!-- Rakuten Web Services Attribution Snippet FROM HERE -->

```

```

<a href="http://webservice.rakuten.co.jp/" target="_blank">Supported by 楽天ウェブサービス</a>
<!-- Rakuten Web Services Attribution Snippet TO HERE -->
  </body>
</html>
</body>
</html>

```

#### 4.3.3.3 商品の削除

商品情報の削除機能は、商品情報削除専用のページにて行う。

構造は、ほしいものリストと同様の商品情報リストを表示し、削除したい商品のチェックボックスにチェックを付けて「削除実行」ボタンをクリックすることで、指定された商品の連番を元に SQL コマンドを利用してデータベース「itemcode」から削除する形をとる。削除の際、削除した商品の連番を表示し、削除しましたというメッセージが表示される。

こちらにも削除完了後ほしいものリストへ戻るためのハイパーリンクを設置する。

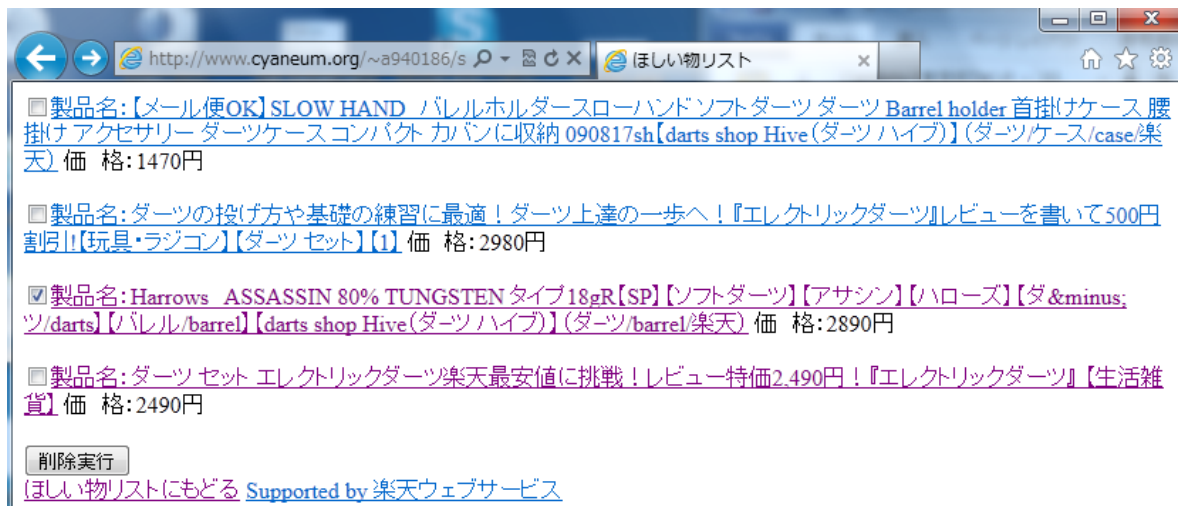


図 4.3.3.3-1 削除ページ

renban	itecode
1	hive:10008712
28	maxshare:10045209
26	hive:10001072
19	smile88:10022071

図 4.3.3.3-2 データベース レコード削除前

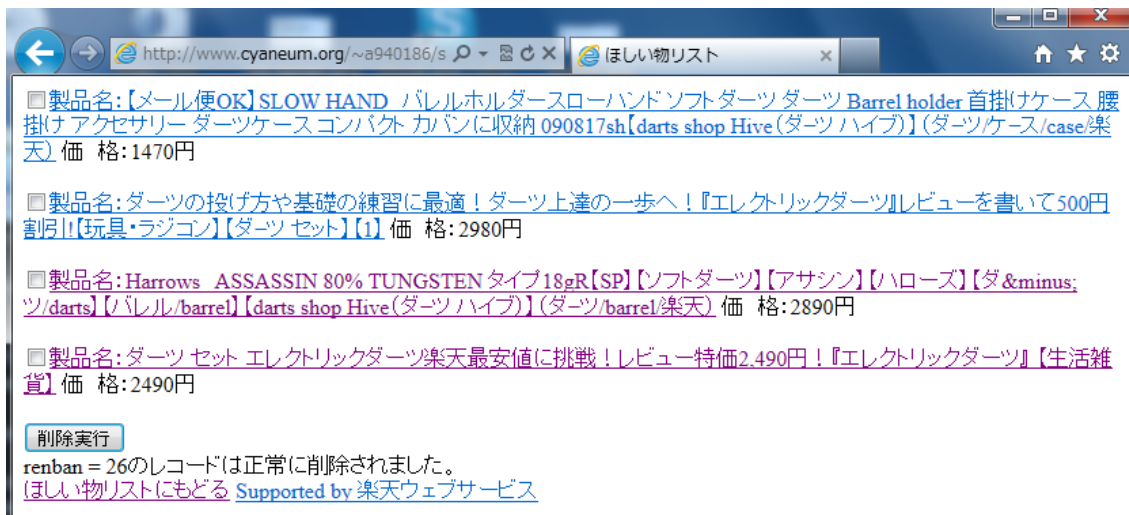


図 4.3.3.3-3 削除ページ 削除実行

renban	itecode
1	hive:10008712
28	maxshare:10045209
19	smile88:10022071

図 4.3.3.3-4 データベース 削除実行結果



## プログラム

このプログラムはデータベース「itemcode」からアイテムコードを抽出し、楽天 API でアイテムコード検索を行うことにより、検索結果をほしいものリストと同様の商品情報リストを表示する。

表示された商品情報の左にはチェックボックスが設置され、チェックボックスにチェックを付けた状態で「削除実行」ボタンをクリックすることにより、削除用のプログラムにチェックを付けた商品の連番を\$renban に代入し、\$renban の数値をもとに指定したレコードを削除する指示を送る。

### sakujo2.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>ほしい物リスト</title>
  </head>
  <body>
<?php

//楽天から商品を検索して HTML タグを返す関数
function search_rakuten($itemcode){

  //echo "itemcode= " . $itemcode . "<br>";
  //developerId を設定する
  $developerId = '*****';

  //affiliateId を設定する
  //$affiliateId = '取得したアフィリエイト ID'; //なくてもよい

  //使用する API の操作名を設定する
```

```
$operation = 'ItemCodeSearch';

//バージョンを設定する
$version = '2010-08-05';

//検索キーワードを URL エンコードして設定する
//$text = urlencode($itemcode);

//URL を生成する
$url = 'http://api.rakuten.co.jp/rws/3.0/rest?'.
    'developerId='.$developerId.
    '&affiliateId='.$affiliateId.
    '&operation='.$operation.
    '&version='.$version.
    '&itemCode='.$itemcode;

//echo $url;
//exit;

//REST で返される XML ファイルを取得する
$str = file_get_contents($url);

//echo $str;
//exit;

$str = str_replace('itemCodeSearch:ItemCodeSearch',
```

```

'itemcodesearch', $str);

//echo $str;

//XML を解析してオブジェクトにセットする
$xml = simplexml_load_string($str);

//var_dump($xml);
//exit;

//xpath によって、Item 要素だけを取り出す
//$Items = $xml->xpath('//Body/itemcodesearch/Item');
$Items = $xml->xpath('//Body/itemcodesearch/Item');
//div タグでくくる
//$ret = '<div id="rakuten_item">';

//Item 数だけ繰り返す
foreach($Items as $Item){
    $ret .= "<a href=¥"". $Item->itemUrl."¥">";
    $ret .= "製品名 : ". htmlspecialchars($Item->itemName) . "</a> ";
    $ret .= "価 格 : ". htmlspecialchars($Item->itemPrice)."円<br/>";
    //$ret .= "店 名 : ". htmlspecialchars($Item->shopName). " </p>";
"<br/>";
    //$zan = $zan-$Item->itemPrice;
}

//div タグを閉じる
//$ret .= '</div>';

```

```

//作成した HTML を返す
return $ret;
}

extract($_POST);

mysql_connect('localhost','a940186','*****');
mysql_select_db('db_a940186');
$sql = "select * from itemcode;";
$result = mysql_query($sql);
$rows = mysql_num_rows($result);
if($rows == 0){
    echo "<p>該当データがありません。 </p>";
}
else {
    echo "<form method='post'>";
    while($row = mysql_fetch_array($result)){
        "<p>";
        echo " <input type='checkbox' name='renban[]' value='".
$row['renban'] ."'>". search_rakuten($row['itecode']) . "<br>";
        "</p>";
    }
    echo "<input type='submit' value='削除実行' />";

    echo "</form>";
}

//チェックした商品を削除する

```

```

$num = count($renban);
if ($num > 0){
    for ($i=0; $i<$num; $i++){
        $sql = "delete from itemcode where renban=" . $renban[$i] . ";";
        $result = mysql_query($sql);
        if (!$result){
            $message = '無効なクエリです。: ' . mysql_error() . "<br>";
            $message .= 'クエリの内容: ' . $sql;
            echo "<p>";
            echo $message;
            echo "</p>";
        } else {
            echo "renban = " . $renban[$i] . "のレコードは正常に削除さ
れました。<br>";
        }
    }
}
?>

<a href = "rist.php">ほしい物リストにもどる</a>

<!-- Rakuten Web Services Attribution Snippet FROM HERE -->
<a href="http://webservice.rakuten.co.jp/" target="_blank">Supported by 楽天ウェブサービス</a>
<!-- Rakuten Web Services Attribution Snippet TO HERE -->
</body>
</html>
</body>

```

</html>

#### 4.3.3.4 残額リセット

データベース「yosan」と「zangaku」、「riyogaku」を更新するためのプログラム。ほしいものリストに設置されている「残額リセット」ボタンを押すことによって更新することができる。また、上限額を変更することも可能にするためボタン横のテキストボックスの数値（最初に入力されているのはデータベース「yosan」に登録されている数値）を変更することによってデータベース「yosan」の数値を更新することができる。

動作としてはまず、データベース「yosan」からデータベース「zangaku」の数値を減算し、今月の利用額を算出する。算出した結果をデータベース「riyogaku」に登録（レコードが保存されている場合は削除を行う）する。そしてデータベース「yosan」、「zangaku」ともにレコードを削除し、テキストボックスから送られてきた数値をPSOTで受け取り、データベース「yosan」と「zangaku」を保存する。

また、完了後ほしいものリストに戻るためのハイパーリンクも設置してある。

この機能は実装に至らなかったためイメージによる解説とする。

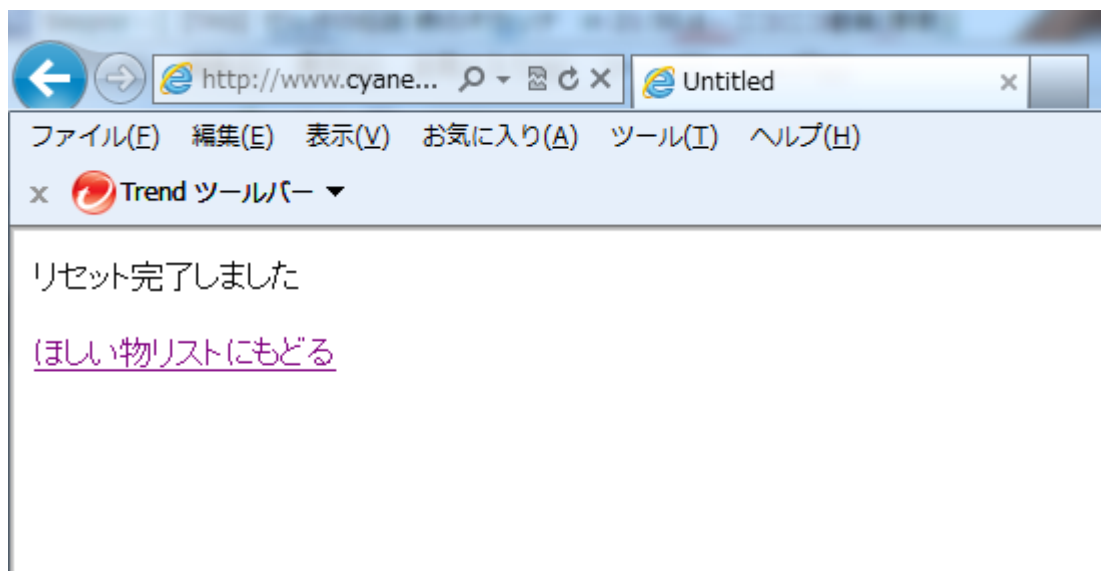


図 4.3.3.4-1 残額リセットの動作

#### プログラムの構想

まずデータベース「yosan」からデータベース「zangaku」の数値を減算し、データベース「riyogaku」に登録の指示を出す。その後データベース「yosan」、「zagaku」

のレコード削除の指示を出し、その後新たに `rist.php` から受け取った数値をデータベース「yosan」、「zagaku」に登録する。

登録後は下部のほしいものリストへのハイパーリンクでほしい物リストへ戻る。

## 5. 評価と考察

### 5.1 評価

本研究によりどれだけの機能が実現されたのか評価をする。

商品検索	商品登録	ほしいものリスト	リストの商品削除
○	○	○	○
計算機能	計算結果反映	リセット	リストの商品選択
○	○	×	○

表 5.1-1 実装評価

表 5.1-1 が本研究での実装状況である。目標としていた機能の内、リセット機能以外の実装は実現した。表 5.1-1 では各機能が実装に至ったかどうかのみの評価になっているため、詳細の評価を行う。

#### ・検索機能

検索機能は、テキストボタンに検索画面のみというシンプルな画面表示で、検索ワードを入力することにより検索を行うことができる。だが、本検索機能は決して便利といえるものではなく、使いづらいものになっている。

その要因としてあげられるのが、

1. 検索に利用できるショッピングサイトが楽天市場のみ
2. ジャンル指定ができない
3. 検索後に表示される商品の数が 5 件のみ
4. 検索ワードによっては出品者違いの同一商品の表示

である。出品者違いの同一商品の表示に関しては、商品の表示件数の少なさに起因するものだが、ジャンル指定ができないことによって、目的のものとは関係のない商品が検索結果に紛れ込んでしまうことがあるため、ただでさえ少ない表示件数を圧迫してしまっている。

また、楽天市場以外のショッピングサイトの API により検索を行うことができるようにすることにより、選択肢の幅が広がることが期待される。

#### ・商品登録

商品登録に関しては現状では特に問題のある動きなどはないが、楽天市場以外のショッピング API 以外の対応を視野に入れる際には、複数のテーブルを選択して商品情報の登録ができるようにする必要がある。



#### ・ほしいものリスト・計算機能・リストの商品選択

ほしいものリスト、計算機能は、一つのページに集約されているためまとめて評価する。ほしいものリスト、計算機能は商品選択の機能も含めて実装はうまくいった。計算機能も複数の商品を選択できるため、一度に複数の商品購入シミュレーションができる。残額を算出した後、残額が 0 以下でないかどうかの判定により、0 以上なら計算結果と計算結果を反映するためのハイパーリンクを表示し、0 以下なら警告文を表示するという動作が実現したため、本研究の金銭管理を支援するシステムの要素の一つを実現できた。

#### ・商品情報削除

削除機能は削除したい商品情報を 1 件ずつ削除するものではなく、専用ページにてチェックボックスを用いて一度に削除することができるものになった。こちらも商品情報をほしいものリストと同じ仕組みでリスト化し商品情報を表示しているため操作しやすいページになった。

#### ・計算結果反映

計算結果反映のページおよびプログラムは動作には問題は見られない。必要な動作をしっかりと行うことができている。

#### ・残額リセット機能

残額のリセット機能に関しては実装できなかったが、実装できなかったことにより、先月の利用額の更新ができず、本研究のシステムの一番重要部分が実現できなかった。残額のリセットだけで言うならば、ほしいものリストの残額のテキストボックスに手動入力することでも代用できるが、利用額の更新を含めたリセット処理はリセット機能でしか実現できないため、この機能が実装できればクレジットカード利用支援システムとしては完成となる。

全体を通して、機能面では概ね実現することができたが、残額リセット機能の実現ができていないことにより、本研究における目標が達成できていないことが第一の問題である。

## 5.2 考察

本研究ではインターネットショッピングでクレジットカードを利用する際、ワンクリックで購入を確定できてしまうために、物を買ったという実感が持ちづらいと感じ、

クレジットカードを利用する際の口座残額や月別の利用額管理を手軽にできないかと考え、学生のインターネットショッピングにおけるクレジットカード利用をより安全に、容易に利用管理ができるようにすることを目標に進めてきた。

研究当初はほしいものリストを作成してその中から選択した商品の購入シミュレーションを行うだけでなく、ショッピングサイトとの何らかの形の連動（ほしいものリストへ追加した商品をショッピングサイトのほしいものリストにも追加するなど）や、複数のショッピング API を利用して複数のショッピングサイトでの検索結果を同時に表示、もしくは検索するショッピングサイトを選択して検索をするなどの機能を考えていたが、現在のようなクレジットカード利用の支援システムとして作成を行った。ショッピングサイトとの連動機能などの実装はできなかったが、ほしいものリストに表示されている商品情報にはその商品ページへのハイパーリンクが設置されており、そのリンクを利用して商品ページに移動してショッピングサイト内のほしいものリストに登録することは可能である。

残額のリセット機能が実装までに至らなかったため、システムとして完全なものを作り上げることができなかった。

結果として本研究ではクレジットカード利用支援システムのうち、ほぼすべての機能が実現できたが、残額のリセット機能が実現しなかったことにより、残額と予算の再設定や、先月の利用額の更新などができないため、今月はあといくら使えるかを見るという限定的な利用方法しかできないものになっている。このため、残額のリセット機能を実装することが第一の課題となった。

本研究での実装がうまく行くと仮定して期待される効果は、ほしいものリストに選択した商品の合計金額や残額、先月の利用額が明記されているため、金銭利用の調節や管理の意識を持つための手助けになると考えられる。また利用額や残額を手動で変更できるためこれをクレジットカードの外部利用分（店舗などでの買い物に利用した際の減額など）に利用することにより、より正確な利用額の算出ができるようになるだろう。

## 6. まとめと今後の課題

### 6.1 まとめ

インターネットショッピングでクレジットカードを利用した際の危機感の持ちづらさを問題視し、買い物に利用した金額をどうにか確認できる形することが必要と考え、ショッピングサイトの種類を問わずに商品をまとめられるほしいものリスト、ほしいものリストへ登録した商品とクレジットカードカードの利用上限額または予算の残額を計算、保存出来る機能を作成したが、実現できたのが楽天市場のみ、また検索の不自由さなど多くの問題が残ってしまった。

また、残額のリセットなどの機能が実現できなかったため、ここで同時に行う予定だった先月の利用額の更新ができなかったため、本研究の目標「自らほしい物と月の利用額の把握を気軽にできるようにし、金銭管理の意識を高めていけるようにすること」のうち、月の利用額の把握がしづらいものになっている。よって本研究の目標は達成できなかった。

だがこのシステムが実現し、検索システムや対応するショッピングサイトを拡張していくことにより、より気軽にネットショッピングにおけるクレジットカードの利用の管理がより容易になり、残額をほしい物リストに常に表示することにより自分がどれほど利用したのかということを確認することができるため、クレジットカード利用の際の限度を意識しやすくなることが期待できる。

### 6.2 今後の課題

今後の課題として、システム完成に至るまでの課題と、完成後の課題が残されている。

#### 完成までの課題

- ・残額リセット機能の実装

残額リセット機能を実装することで、月ごとに残額のリセットを行うことができるようになり月別の利用額を明確にして、ほしいものリストに先月の利用額として表示することにより、より容易に金銭管理を行うことができるようにする。またこの機能により、予算額の変更を行えるようになるため、より月別の管理に自由度を持たせることができる。

本研究での実装予定は、基本的な利用を可能にする程度にとどまっていたため、快

適に利用するために以下のような改善、拡張を加えることが必要であると考えます。

- ・ 検索機能の拡張

検索機能は現在楽天市場の商品検索しか利用できない。ほしい物をより安く、または自由に選択できるよう amazon や yahoo ショッピングなど他のショッピング API を利用して検索に利用できるショッピングサイトを増やす。

また、ジャンル検索の実装、また表示数の増加などにより見やすい検索ページにすることができる。

- ・ 商品の購入判定

本研究で作成していたシステムは、計算結果を反映した際に変更されるのは残額だけであり、ほしいものリストにも商品情報が残ってしまい、購入シミュレーションした後に自分で商品情報を削除しなければならない。購入判定を商品情報に追加し、購入済みページを作成出来ればより一層金銭管理が容易になると考えられる。

本研究での実装の課題は、まずはシステム自体を完成させるという点が一番の課題であるが、完成後の問題点を改善出来ればシステムとしての使いやすさは格段に向上することが期待される。

## 参考文献

- [1]amazon.co.jp(<http://www.amazon.co.jp/>) 平成 24 年 11 月 20 日
- [2]PHPBook(<http://www.phpbook.jp/>) 平成 24 年 11 月 24 日
- [3]PHP Labo(<http://www.php-labo.net/>) 平成 24 年 12 月 3 日
- [4]PHP マニュアル(<http://php.net/manual/ja/index.php>) 平成 24 年 12 月 3 日
- [5]PHP プロ(<http://www.phppro.jp/>) 平成 24 年 12 月 3 日
- [6]V プリカ(<http://vpc.lifecard.co.jp/>) 平成 24 年 12 月 10 日
- [7]WebMoney(<http://www.webmoney.jp/>) 平成 24 年 11 月 15 日
- [8]いいめも(<http://www.ememo.jp/okodukai/>) 平成 24 年 12 月 10 日
- [9]独立行政法人日本学生支援機構 平成 22 年度学生生活調査について  
([http://www.jasso.go.jp/tokubetsu\\_shien/chosa1101.html](http://www.jasso.go.jp/tokubetsu_shien/chosa1101.html)) 平成 25 年 1 月 9 日
- [10]楽天市場(<http://www.rakuten.co.jp/>) 平成 24 年 11 月 20 日

## 謝辞

本研究を行うに当たり、ご指導を頂いた渡辺 恭人准教授に心より感謝いたします。論文作成上での適切なアドバイスや、私が作成に手間取っていたプログラムの設計を理解していただき、細かな質問に対しても熱心に指導頂いたお蔭で、本来予定していた以上の成果を上げることができました。

2年生の頃からゼミに参加させていただき、プログラミング初心者の私に1から指導して頂いたお蔭で、まだまだ未熟ではありますが、問題点を発見し研究を行い、ここまで研究し成果を作り上げることができました。

最後に、私の論文に関わってくださった方全員に感謝の意を述べさせていただき、謝辞といたします。本当にありがとうございました。