

「コーディネート支援のためのアイテム提案機能の検討」

指導教員：渡辺恭人

1040005

河村美咲

提出日：2013年12月16日

## 目次

1. 背景・目的.....	4
1-1 背景	
1-2 目的	
2. 現状と問題点.....	5
2-1 コーディネートの現状	
2-1-1 ヒアリング調査	
2-1-2 既存のサービス	
2-1-3 丸山さんの卒業研究	
2-1-4 雑誌	
2-2 問題点	
3. 解決法の検討.....	12
3-1 解決法のアプローチ	
3-2 必要な機能の提案	
3-2-1 持っている服の管理機能	
3-2-2 過去のコーディネートの記録機能	
3-2-3 ソーシャル化機能	
3-2-4 アイテムの提案機能	
4. 実装と設計.....	15
4-1 アプリケーション設計	
4-1-1 全体イメージ	
4-1-2 アイテムページ	
4-1-3 コーディネート登録ページ	
4-2 データベーステーブル	
4-2-1 アイテム登録時のアイテム情報テーブル	
4-2-2 コーディネート登録時のアイテム情報テーブル	
4-2-3 カテゴリテーブル	

4-2-4	天気テーブル	
4-2-5	TPO テーブル	
4-2-6	コーディネートテーブル	
4-3	実装	
4-3-1	実装環境	
4-3-2	アイテム登録	
4-3-3	アイテム検索	
4-3-4	コーディネート作成	
4-3-5	コーディネート登録	
5.	評価と考察.....	56
5-1	動作確認	
5-1-1	トップページ	
5-1-2	アイテム登録ページ	
5-1-3	アイテム検索ページ	
5-1-4	アイテム削除ページ	
5-1-5	コーディネート作成・登録ページ	
5-2	機能評価	
5-3	考察	
6.	まとめと今後の課題.....	66
6-1	まとめ	
6-2	今後の課題	
	参考文献.....	68
	謝辞.....	69

## 1. 背景・目的

### 1-1 背景

服は毎日着るものだから毎日のコーディネートで悩む人はたくさんいる。特に多くの女性は、様々な服を着てコーディネートを楽しむ為に毎日時間をかけて考えている。さらに、その中で曜日、その日に会う人、その日の気分、天気などコーディネートの仕方は人それぞれである。そして毎日コーディネートに時間がかかってしまう人もたくさんいる。実際に筆者(河村)も曜日や会う人、天気など様々なコーディネートの仕方をしている。また、せっかくだくさんの服を所持しているのにも関わらず、最近着た服ばかりを着て以前購入した服の存在を忘れている場合もある。その為、いつでも自分の服情報がわかり、コーディネートを考えることできたら時間短縮やコーディネートを考える時の煩わしさが軽減する。

本研究では、誰でも簡単にいつでもどこでもコーディネートをできる仕組みを提案する。

### 1-2 目的

持っている服を把握し、管理すると共に過去のコーディネート履歴を管理することで毎日のコーディネートの時間短縮に繋げる。それにより、クローゼットの中をいちいち見なくても簡単に持っている服の確認ができ、手間を省くことができる。また、管理している服とコーディネート情報からアイテムを提案し、コーディネートを決める時の更なる時間短縮と手間を省くことを目的とする。

さらに、他の人とコーディネート情報を共有することでコーディネートの質を向上させることができる。コーディネート情報を共有することで自分のコーディネートを評価してもらったり、他の人のコーディネートを見たりすることでコーディネートの幅を広げ、さらにコーディネートを楽しむことができる。

## 2. 現状と問題点

### 2-1 コーディネートの現状

この節では、コーディネートについて様々な事前調査をした結果について述べる。

#### 2-1-1 ヒアリング調査

この項では事前にヒアリング調査をした結果について述べる。ヒアリング調査の対象と質問項目を表 2-1 に示す。

表 2-1：ヒアリング調査の詳細

対象	女子大学生、社会人女性
人数	7人(女子大学生4人、社会人女性3人)
質問項目	①コーディネートを決める時に気にしていること と ②同じ服を着用するのを避ける期間

#### ①コーディネートを決める時に気にしていること

コーディネートを決める時に最も気にするのは、新しく購入した服の着回しだった。その次に、天気だった。しかし、新しく購入した服を着回しする人が多数な半面、同じ服を着ることを避けることも重要だということがわかった。新しく購入した服を着回しするが同じコーディネートを避けている人は、新しく購入した服を何度も着るが、他に合わせる服を変えることで同じ服だけ雰囲気を変えていた。

また、靴やアクセサリなどの小物から決める人もいることがわかった。

#### ②同じ服を着ることを避ける期間

同じ服を着ることを避ける期間は人によって様々で、全員1日～2週間の間、同じ服を着ることを避けていた。①で述べた通り、一つの服を何度も着ることもあるが他の服を変えて違うコーディネートをしている人もいた。また、連日同じ人に会わなければ、同じ服を着るとい人もいた。この為、同じ服を着ることをさける期間は一概に〇日だとは言えない。

## 2-1-2 既存のサービス

### (1) おしゃれ天気

おしゃれ天気は、天気予報と共にこの日の天気や気温に合ったコーディネートを表示してくれるスマートフォン用のアプリケーション（iPhone、Android 用）である。

Twitter と連動させて天気を簡単にフィードすることもできる。

天気予報とコーディネートを表示する画面を図 2-1-3-1 に示す。



図 2-1-3-1：おしゃれ天気

機能：天気予報閲覧、コーディネートの提案

メリット：

- ・天気とその日の天気合ったコーディネートと一緒に閲覧できる。
- ・その日の天気や気温が1時間、3時間ごとに見ることができる。
- ・毎日、設定した時間にその日の天気と気温のお知らせが届く。
- ・コーディネートを何種類か提案してくれる。
- ・天気予報を表示するエリアを5件まで登録できる。

デメリット：

- ・提案されたコーディネートのような服を自分が持っていない場合がある。
- ・今日と明日のコーディネートしか提案されない。

使用した感想

毎朝、天気と気温のお知らせが届き、そこからアプリケーションをすぐに起動するのが良かった。また、その日の天気合ったコーディネートをイラストで表示してくれるので自分が同じような服を持っていなくてもイメージしやすかった。何日か先

のコーディネートも見られると良い。

## (2) CoordiSnap

CoordiSnap は、自分のお気に入りのコーディネートやアイテムをコメントと共に投稿して、閲覧者から評価をもらうことができるスマートフォン用のアプリケーション（iPhone、Android 用）である。

現在、評価が高い人気のコーディネートを表示する画面を図 2-1-3-2 に示す。



図 2-1-3-2 : CoordiSnap

機能：コーディネートへの投稿、閲覧、検索、ワンクリック評価、コメント記入、お気に入り登録

メリット：

- ・ワンクリック評価があるので良いコーディネートが一目でわかる。
- ・検索で自分の着たいコーディネートが検索できる。

デメリット：

- ・このアプリケーションを持っている人全員に閲覧される。
- ・検索しても自分の着たいコーディネートがあるとは限らない。
- ・コーディネート画像の表示画面が見づらい。

使用した感想

ワンクリック評価があり、評価数を見ると一目で人気のコーディネートがわかるので良かった。また、一言コメントもできるので他のユーザにコーディネートのどこが良かったのか具体的にコメントしてもらえ、一方、他のユーザのコーディネートにコ

メントで服の購入場所などを質問できたのも良かった。しかし、検索機能はキーワード検索のみで自分の好みのコーディネートを探すことが難しかった。

### (3) FUKULOG

FUKULOG は、世界中のコーディネートやアイテムなどのファッション情報を投稿したり、閲覧したり、好みと同じ人たちと交流したりするソーシャルファッションサイトである。パソコン、スマートフォンからの閲覧が可能である。

新着のコーディネートやアイテムの一覧を表示する画面を図 2-1-3-3 に示す。

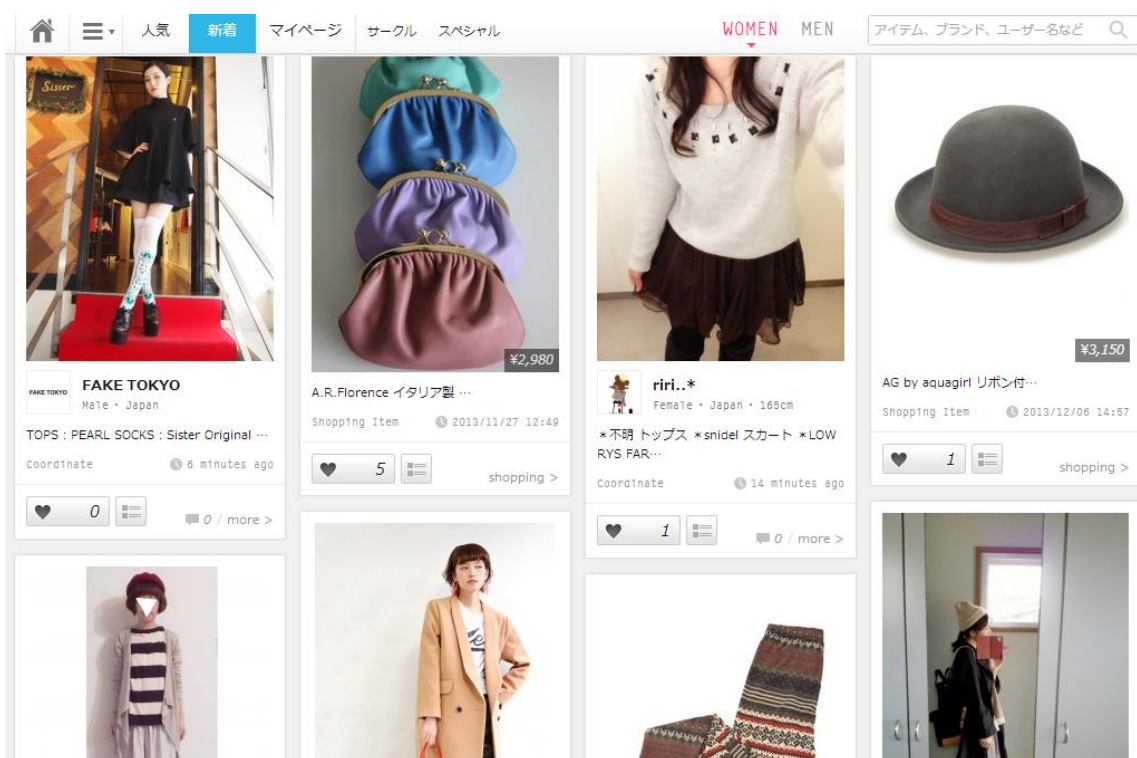


図 2-1-3-3 : FUKULOG

機能：コーディネートへの投稿、閲覧、検索、ワンクリック評価、コメント記入、お気に入り登録、サークルでの交流、ユーザ同士のメッセージ交換、ブランドやショップからのニュースの閲覧

メリット：

- ・ワンクリック評価があるので良いコーディネートが一目でわかる。
- ・検索で自分の着たいコーディネートが検索できる。
- ・ユーザ同士のメッセージ交換ができたり、同じ年代や趣味趣向を持ったユーザ同士でサークルを作って交流できたりするのでコミュニケーションを取りながら情報の交換・共有ができる。



- ・ランドやショップからのニュースの閲覧でファッションの最新情報を入手できる。

デメリット：

- ・コーディネート提案はしているが、自分の所持している服の管理はできないので、投稿している人たちの服を実際に自分が所持していないことが多い。

#### (4) suGATALOG

suGATALOGは、実際に服を着て写真を撮り、その写真を組み合わせてコーディネートを作成していくアプリケーションで、パソコンとスマートフォン（iPhoneのみ）で利用できる。

スマートフォンを利用してコーディネートを作成している画面を図2-1-3-4に示す。



図 2-1-3-4： suGATALOG

機能：コーディネートの作成、閲覧、お気に入り登録

メリット：

- ・実際に服を着用した写真を使用するので、着用イメージがわかりやすい。
- ・オートコーディネート機能があり、好きなアイテムに合うコーディネートを提案してくれる。
- ・お気に入り登録しているコーディネートを一覧で並べて閲覧できる。

デメリット：

- ・所持している全ての服を着用して写真を撮らなくてはならないので手間がかかる。
- ・組み合わせできるのはトップス、ボトムスだけなので小物や靴を合わせたイメージができない。

#### (5) 既存のサービスの比較

2-1-3-1 から 2-1-3-4 で述べた 4 つのサービスについて、アイテム管理、コーディネート提案、コーディネート評価の点で比較したものを図 2-1-3-5 に示す。

	おしゃれ天気	CoordiSnap	FUKULOG	suGATALOG
アイテム管理	×	×	×	△ トップス・ボトムスのみ
コーディネート提案	○	×	×	○
コーディネート評価	×	○	○	×
ソーシャル化	×	○	○	×

図 2-1-3-5：既存のサービスの比較

#### 2-1-3 丸山さんの卒業研究

本項では、2012 年度卒業生の丸山みゆきさんの卒業研究「コーディネート支援のための所有衣服管理とアイテム提案機能の検討」を基に述べる。

##### ・アイテム登録

カテゴリ、カテゴリ内訳、色、柄の有無、季節などアイテムを登録する際にたくさんの情報を入力している。その為、アイテムの詳細データを管理できる。

##### ・アイテム表示

全てのアイテムを登録順に表示している。しかし、並べ替え機能が実装できていなかった為、閲覧したいアイテム情報を表示するのを時間がかかってしまうことがある。

##### ・コーディネート登録

カテゴリからアイテムを選択してコーディネートを作成して登録している。アイテムの選択方法はカテゴリから選択していく方法のみで、コーディネート登録時にアイ

テムの着用履歴は更新されない。

- ・ 除外機能

近日中に着用したアイテムを 3 日間と 1 週間避けることができていた。本研究でも除外機能を取り入れていきたい。

#### 2-1-4 雑誌

ファッション雑誌では、よく着回してコーディネートをする企画を掲載している。宝島社の mini というファッション雑誌では、着回し企画を 2013 年では、3 月号と 8 月号の約半年に 1 回のペースで掲載されていた。この企画では同じ服を着ることをあまり避けていないが、同じ服でも服の組み合わせ次第でコーディネートの印象が変わることがわかった。

2-1-1 で新しく購入した服を着回しするが同じ服を着ることを避けている人の中には、雑誌の着回しを参考にしてコーディネートを考えている人もいた。

#### 2-2 問題点

コーディネートの決め方は人それぞれであり、同じ人でも日によって決める基準は変わる。また、同じ服を避ける期間も人それぞれである。その中で同じコーディネートは避けるが、同じ服は短期間で着てコーディネートを変えて印象を変える人もいる。

よって、コーディネートを決める方法と同じ服を避ける期間は、選択できるようにしたほうが良い。

既存のサービスだと個々のサービスの重視している機能は良い為、少しずつ取り入れていきたいが、実現したい機能が全て備わっているものがない。また、丸山さんの研究では、アイテム管理を中心としてそこからコーディネート選択をしているので、提案という点では改善が必要である。しかし、アイテム管理やコーディネート作成時の除外機能などは優れているので取り入れていく。

### 3. 解決法の検討

#### 3-1 解決法のアプローチ

所有している服の情報と過去のコーディネート情報をただ管理していても、その管理している情報の中から自分で探し出してコーディネートを作成するには時間がかかってしまう。コーディネートを作成する時間を短縮するには、アイテムを提案してくれる仕組みを構築することが重要である。また、アイテムの提案は、2-2節の問題点で述べた通り、コーディネートを決める方法と同じ服を避ける期間は人それぞれである為、コーディネートを決める方法と同じ服を避ける期間は選択できるようにしたほうが良い。

さらに、ソーシャル化をして他の人とコーディネートの情報を共有して評価してもらったり、新しいコーディネートの情報を取り入れたりすることでコーディネートを楽しむことを目指す。

#### 3-2 必要な機能の提案

##### 3-2-1 持っている服の管理機能

持っている服を登録・閲覧・削除ができる機能。アイテムの提案をする為に服のカテゴリ、天気、TPO、購入日、最終着用日、着用回数などの情報を登録して管理する必要がある。また、写真も一緒に登録することで、どの服の情報かが一目でわかるのでコーディネートを決めやすくなる。

##### 3-2-2 過去のコーディネートの記録機能

過去のコーディネートを記録して、履歴を見ることができる機能。同じコーディネートを避けたり、コーディネートを決める時に選択の1つにしたりする。2-1-2項で述べた通り、コーディネートを決める時には同じコーディネートを避ける人が多いので、過去のコーディネートを記録するのは必要である。

##### 3-2-3 ソーシャル化機能

図 3-2-3 のようにコーディネートを投稿し、他のユーザから「いいね！」のようなワンクリック評価をもらう。さらにその評価をコーディネート情報に反映させ、コーディネートを決める時に役立てる。また、他のユーザが投稿したコーディネートを閲

覧して、新たなコーディネート情報を収集し、コーディネートを楽しむことができる。



図 3-2-3：ワンクリック評価の一例（CoordiSnap より引用）

#### 3-2-4 アイテムの提案機能

天気予報・トップスから・ボトムスから・好きな基準（最近買った服、最近よく着ている服、最近来てない服など）・TPO・コーディネート履歴の 6 つのアイテム選択方法からその日に合わせて自分で好きな選択方法を決定し、そこからアイテムを提案していく。さらに、アイテム除外機能で近日中に着た服を除外してアイテムを提案する。また、アイテムを除外する期間は好きなように選択できるようにする。アイテムの提案機能の仕組みのイメージを図 3-2-4 に示す。

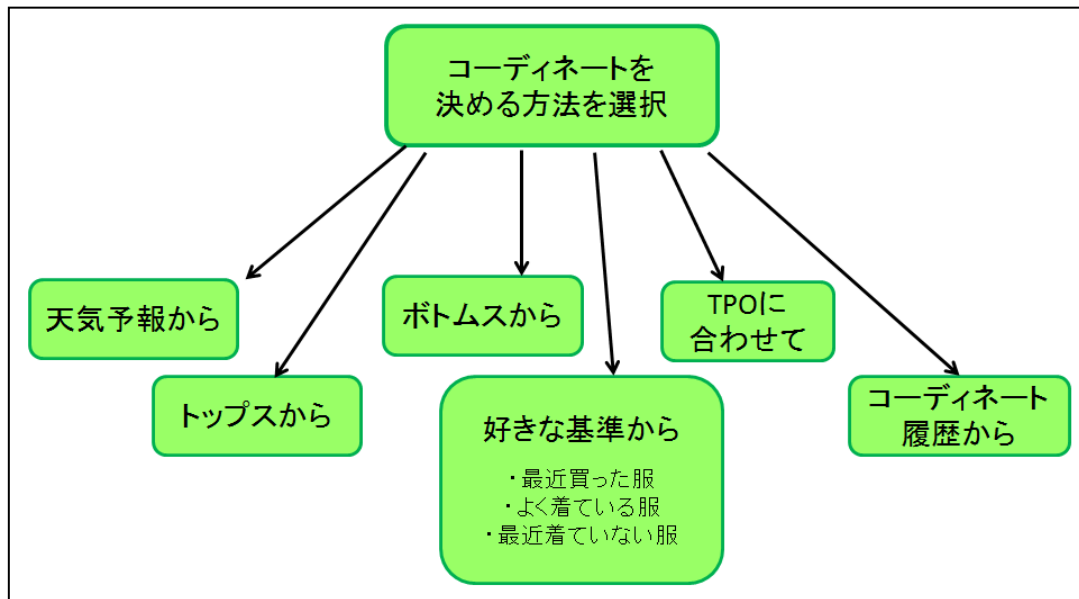


図 3-2-4 : アイテム提案機能のイメージ

## 4. 設計と実装

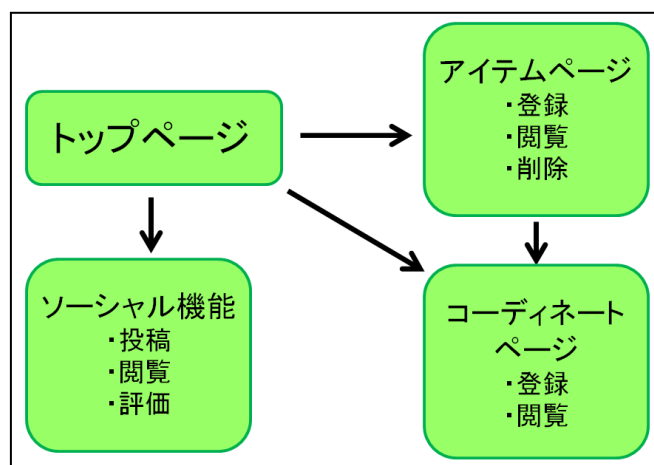
### 4-1 アプリケーション設計

#### 4-1-1 全体イメージ

トップページからアイテムページ、ソーシャル機能ページの各ページへの移動ができる。コーディネートページへは、アイテムページから移動できる。

トップページのイメージを図 4-1-1-1 に表す。

図 4-1-1-1：トップページのイメージ



#### 4-1-2 アイテムページ

アイテムページでは、新しいアイテムの登録、コーディネートページ、現在所有しているアイテムの閲覧、いらなくなったアイテムの削除ができる。また、アイテムページのアイテムの閲覧からもコーディネート登録ページへの移動ができるようにする。

アイテムページのイメージを図 4-1-2-1 に表す。

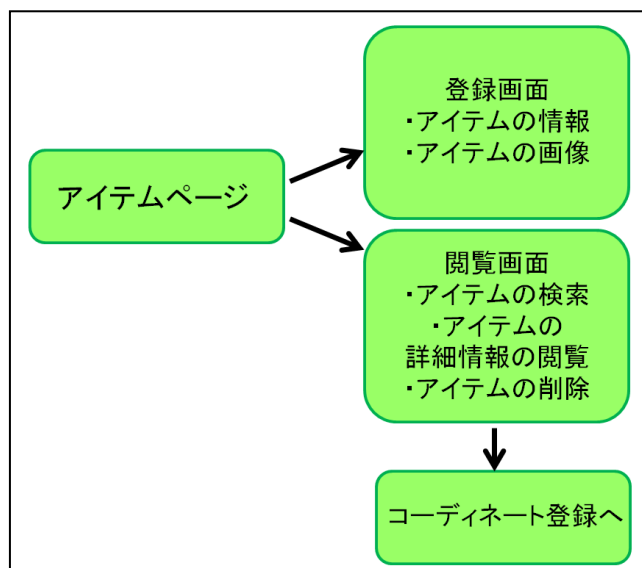


図 4-1-2-1 : アイテムページのイメージ

#### 4-1-3 コーディネート登録ページ

コーディネート登録ページは、トップページとアイテムページの両方から移動できる。コーディネート登録はアイテムページで検索したアイテムを反映させる。

コーディネート登録ページのイメージは図 4-1-3-1 に表す。

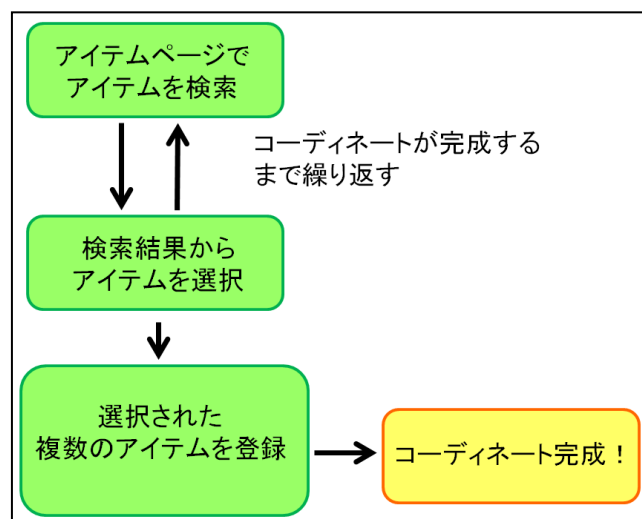


図 4-1-3-1 : コーディネート登録ページのイメージ



## 4-2 データベーステーブル

### 4-2-1 アイテム登録時のアイテム情報テーブル

所持している服の情報を管理する為のテーブルを作成する。このデータベースのテーブル設計を次の表 4-2-1-1 に示す。

表 4-2-1-1 : テーブル (fuku)

項目内容	フィールド名	型
番号	id	int, auto-increment, index
購入日	kounyu	date
カテゴリ	cate	int
天気	weather	int
TPO	tpo	int
画像	pic	char(20)
備考	biko	text

表 4-2-1-1 を基に次の表 4-2-1-2 でテーブルを作成する。服の情報は後から追加することがある為、id を連番の設定をする。

表 4-2-1-2 : テーブル fuku 作成

```
mysql> create table fuku (id int, kounyu date, cate int, weather int, tpo
int, pic char(20), biko text);
Query OK, 0 rows affected (0.00 sec)

mysql> alter table fuku add index ind1(id);
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table fuku modify id int auto_increment;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show fields from fuku;
```

```
+-----+-----+-----+-----+-----+-----+
| Field  | Type      | Null  | Key  | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id     | int(11)   | NO    | MUL  | NULL    | auto_increment |
| kounyu | date      | YES   |      | NULL    |              |
| cate   | int(11)   | YES   |      | NULL    |              |
| weather | int(11)   | YES   |      | NULL    |              |
| tpo    | int(11)   | YES   |      | NULL    |              |
| pic    | char(20)  | YES   |      | NULL    |              |
| biko   | text      | YES   |      | NULL    |              |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

#### 4-2-2 コーディネート登録時のアイテム情報テーブル

コーディネート登録時に所持している服の情報に追加される情報を管理する為のテーブルを作成する。このデータベースのテーブル設計を次の表 4-2-2-1 に示す。

表 4-2-2-1 : テーブル (fuku\_sub)

項目内容	フィールド名	型
番号	id	int
回数	kaisu	int
最終着用日	saigo	date

表 4-2-2-1 を基に次の表 4-2-2-2 でテーブルを作成する。服の情報は後から追加することがある為、idを連番の設定をする。

表 4-2-2-2 : テーブル fuku\_sub 作成

```
mysql> create table fuku_sub (id int, kaisu int, saigo date);
```

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

```
mysql> show fields from fuku_sub;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)   | YES  |     | NULL    |       |
| kaisu | int(11)   | YES  |     | NULL    |       |
| saigo | date      | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

```
3 rows in set (0.01 sec)
```

#### 4-2-3 カテゴリテーブル

テーブル fuku で必要となるカテゴリのテーブルを作成する。このデータベースのテーブル設計を次の表 4-2-3-1 に示す。

表 4-2-3-1 : テーブル (cate)

項目内容	フィールド名	型
カテゴリ番号	c_id	int
名称(トッパス、ボトムス…など)	c_name	varchar(30)

表 4-2-3-1 を基に次の表 4-2-3-2 でテーブルを作成する。

表 4-2-3-2 : テーブル cate 作成

```
mysql> create table cate (c_id int, c_name varchar(30));
```

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

```
mysql> show fields from cate;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
```

```

+-----+-----+-----+-----+-----+
| c_id  | int(11)  | YES  |      | NULL  |      |
| c_name| varchar(30)| YES  |      | NULL  |      |
+-----+-----+-----+-----+

```

2 rows in set (0.00 sec)

```
mysql> insert into cate values (1, 'Tops');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into cate values (2, 'Bottoms');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into cate values (3, 'OnePiece');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into cate values (4, 'Outer');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into cate values (5, 'Shoes');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into cate values (6, 'Accessory');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> select * from cate;
```

```

+-----+-----+
| c_id | c_name  |
+-----+-----+
|    1 | Tops    |
|    2 | Bottoms|

```

```

| 3 | OnePiece |
| 4 | Outer    |
| 5 | Shoes    |
| 6 | Accessory|
+-----+-----+
6 rows in set (0.00 sec)

```

#### 4-2-4 天気テーブル

テーブル fuku で必要となる天気のテーブルを作成する。このデータベースのテーブル設計を次の表 4-2-4-1 に示す。

表 4-2-4-1：テーブル (weather)

項目内容	フィールド名	型
天気番号	w_id	int
名称 (雨、晴れ…など)	w_name	varchar(30)

表 4-2-4-1 を基に次の表 4-2-4-2 でテーブルを作成する。

表 4-2-4-2：テーブル weather 作成

```

mysql> create table weather (w_id int, w_name varchar(30));
Query OK, 0 rows affected (0.00 sec)

mysql> show fields from weather;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| w_id  | int(11)       | YES  |     | NULL    |       |
| w_name| varchar(30)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

```

mysql> insert into weather values (1, 'Hare');
Query OK, 1 row affected (0.00 sec)

mysql> insert into weather values (2, 'Ame');
Query OK, 1 row affected (0.00 sec)

mysql> insert into weather values (3, 'Kumori');
Query OK, 1 row affected (0.00 sec)

mysql> insert into weather values (4, 'Yuki');
Query OK, 1 row affected (0.00 sec)

mysql> insert into weather values (5, 'Nashi');
Query OK, 1 row affected (0.00 sec)

mysql> select * from weather;
+-----+-----+
| w_id | w_name |
+-----+-----+
|    1 | Hare   |
|    2 | Ame    |
|    3 | Kumori |
|    4 | Yuki   |
|    5 | Nashi  |
+-----+-----+
5 rows in set (0.00 sec)

```

#### 4-2-5 TPO テーブル

テーブル fuku で必要となる TPO のテーブルを作成する。このデータベースのテーブ

ル設計を次の表 4-2-5-1 に示す。

表 4-2-5-1 : テーブル (tpo)

項目内容	フィールド名	型
TPO 番号	t_id	int
名称 (仕事、正装…など)	t_name	varchar(30)

表 4-2-5-1 を基に次の表 4-2-5-2 でテーブルを作成する。

表 4-2-5-2 : テーブル tpo 作成

```
mysql> create table tpo (t_id int, t_name varchar(30));
Query OK, 0 rows affected (0.00 sec)

mysql> show fields from tpo;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| t_id  | int(11)       | YES  |     | NULL    |      |
| t_name | varchar(30)   | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+

2 rows in set (0.00 sec)

mysql> insert into tpo values (1, 'Work');
Query OK, 1 row affected (0.00 sec)

mysql> insert into tpo values (2, 'Formal');
Query OK, 1 row affected (0.00 sec)

mysql> insert into tpo values (2, 'Sports');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into tpo values (3, 'Sports');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into tpo values (4, 'Date');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into tpo values (5, 'Other');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into tpo values (6, 'Nashi');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from tpo;
```

```
+-----+-----+
```

```
| t_id | t_name |
```

```
+-----+-----+
```

```
| 1 | Work |
```

```
| 2 | Formal |
```

```
| 3 | Sports |
```

```
| 4 | Date |
```

```
| 5 | Other |
```

```
| 6 | Nashi |
```

```
+-----+-----+
```

```
6 rows in set (0.00 sec)
```

#### 4-2-6 コーディネートテーブル

登録されたコーディネート进行管理する為のデータベーステーブルを作成する。このデータベースのテーブル設計を次の表 4-2-6-1 に示す。

表 4-2-6-1 : テーブル (coordinate)



項目内容	フィールド名	型
コーディネート番号	cd_id	int
着用日	chaku	date
トップス ID	cate1	int
ボトムス ID	cate2	int
ワンピース ID	cate3	int
アウターID	cate4	int
靴 ID	cate5	int
小物 ID	cate6	int

表 4-2-6-1 を基に次の表 4-2-6-2 でテーブルを作成する。

表 4-2-6-2 テーブル coordinate 作成

```
mysql> create table coordinate(cd_id int, chaku date, cate1 int, cate2 int, cate3
int, cate4 int, cate5 int, cate6 int);
Query OK, 0 rows affected (0.00 sec)

mysql> alter table coordinate add index ind1(cd_id);
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table coordinate modify cd_id int auto_increment;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show fields from coordinate;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| cd_id | int(11) | NO   | MUL | NULL    | auto_increment |
```

chaku	date	YES		NULL	
cate1	int(11)	YES		NULL	
cate2	int(11)	YES		NULL	
cate3	int(11)	YES		NULL	
cate4	int(11)	YES		NULL	
cate5	int(11)	YES		NULL	
cate6	int(11)	YES		NULL	
+-----+-----+-----+-----+-----+-----+					
8 rows in set (0.00 sec)					

### 4-3 実装

#### 4-3-1 実装環境

本研究は以下の実装環境で行った。

##### (1) クライアント環境

オペレーティングシステム：Windows Vista Home Premium

CPU：Intel(R) Core(TM)2 Duo CPU P8600 @2.4GHz 2.40GHz

メモリ：2.00GB

##### (2) サーバ環境(学外：www.cyaneum.org)

オペレーティングシステム：Ubuntu 8.04 server

CPU：Pentium M 1.6MHz

メモリ：

Webサーバ：Apache 2.2.8

データベース：MySQL5.0.96

##### (3) プログラミング言語

PHP(Version 5.2.4-2)とHTML

#### 4-3-2 アイテム登録

所持している服の情報を登録する為のデータの流れを図 4-3-2-1 に示し、プログラムを表 4-3-2-2 と表 4-3-2-3 に示す。

このプログラムでは、購入日、カテゴリ、天気、TPO、画像、備考の6つの情報をデ

データベース fuku に登録している。カテゴリ、天気、TPO はプルダウンメニューから選択できるようになっている。また、データベース fuku に登録が完了するとともに id、着用回数、最終着用日の情報がデータベース fuku\_sub に登録される。着用回数、最終着用日には 0 が入力される。

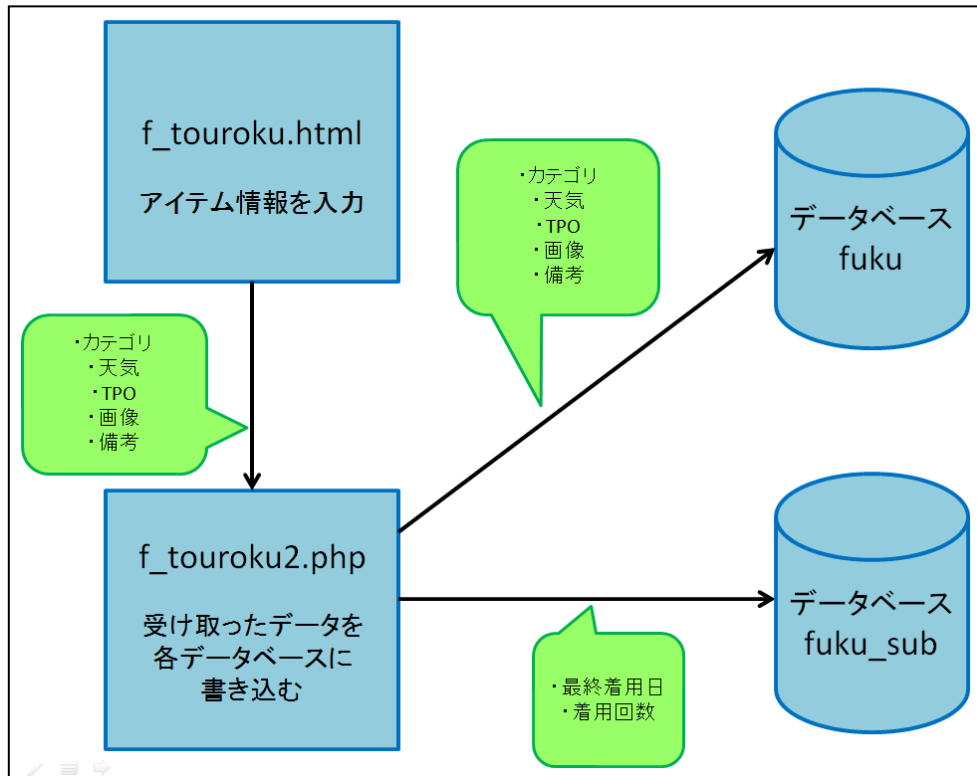


図 4-3-2-1：アイテム登録のデータの流れ

表 4-3-2-2：f\_touroku.html

```

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

<html lang="ja">
<head>
  <title>服情報登録</title>
  <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8">
</head>
<body>
  
```

```
<p><font size="5">*データを入力*</font></p>
<form action="f_touroku2.php" method="post" enctype="multipart/form-data">
<p>購入日 : <input type="text" name="nen" size="10">
      -<input type="text" name="tuki" size="5">
      -<input type="text" name="niti" size="5"></p>
<p>カテゴリ : <select name="cate" size="1">
      <option value="1">トップス</option>
      <option value="2">ボトムス</option>
      <option value="3">ワンピース</option>
      <option value="4">アウター</option>
      <option value="5">靴</option>
      <option value="6">小物</option>
      </select></p>
<p>天気 : <select name="ten" size="1">
      <option value="1">晴れ</option>
      <option value="2">雨</option>
      <option value="3">曇り</option>
      <option value="4">雪</option>
      <option value="5">指定なし</option>
      </select></p>
<p>TPO : <select name="tpo" size="1">
      <option value="1">仕事</option>
      <option value="2">正装</option>
      <option value="3">運動</option>
      <option value="4">デート</option>
      <option value="5">その他</option>
      <option value="6">指定なし</option>
      </select></p>
<p>画像 : <input type="file" name="gazo" size="30"></p>
<p>備考 : <textarea name="bik" rows="10" cols="50">
```

```

</textarea></p>
<p><input type="submit" value="登録">
<input type="reset" value="リセット"></p>
</form>
</body>
</html>

```

表 4-3-2-3 : f\_touroku2. php

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
  <head>
    <title>*データ入力*</title>
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
  </head>
  <body>
<?php
extract($_POST);
mysql_connect('localhost', 'b04****', '*****');
mysql_select_db('b04****');

$img = $_FILES[ "gazo" ][ "tmp_name" ];
$gazo = $_FILES[ "gazo" ][ "name" ];

$save_directory = "pic/";
$path = realpath($save_directory);

if($gazo <>''){
    if ( $_FILES[ 'gazo' ][ 'error' ] == UPLOAD_ERR_OK &&

```

```

is_uploaded_file( $img ) ){
    move_uploaded_file( $img, "pic/" . $gazo );
    chmod("pic/" . $gazo, 0644);
    echo "アップロード成功!";
} else {
    echo "アップロード失敗!";
}
$jun = "$nen-$tuki-$niti";
$sql = "insert into fuku values(0, '$jun', '$cate', '$ten', '$tpo', '$gazo',
'$bik')";
$result = mysql_query($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 {
    $sql = "insert into fuku_sub values((select max(id) from fuku), 0, 0)";
    mysql_query($sql);
    echo "正常に登録されました!<br>";
    echo "<a href=f_touroku.html>登録ページに戻る</a>";
}
}
?>

</body>

</html>

```

登録された服の情報を表示と削除する為のデータの流れを図 4-3-3-1 に示し、プログラムを表 4-3-3-2 に示す。データベース fuku と連動している。

チェックボタンでカテゴリ、天気、TPO を絞って検索することができる。また、近日中に着た服を除外させる期間と表示順をプルダウンメニューで選択できる。選択できる除外期間は 3 日。1 週間、2 週間で除外しないこともできる。選択できる表示順は着用回数の少ない順、着用回数の多い順、着用日の昇順、着用日の降順、購入日の昇順、購入日の降順である。このプログラムでは id、画像、購入日、カテゴリ、天気、TPO、着用回数、最終着用日、備考が表示される。着用回数が 0 回の場合、最終着用日は「未着用」と表示させる。

さらに、検索結果で表示された服からコーディネートを作成と服の削除ができる。削除の為のプログラムは表 4-3-3-3 に表示する。

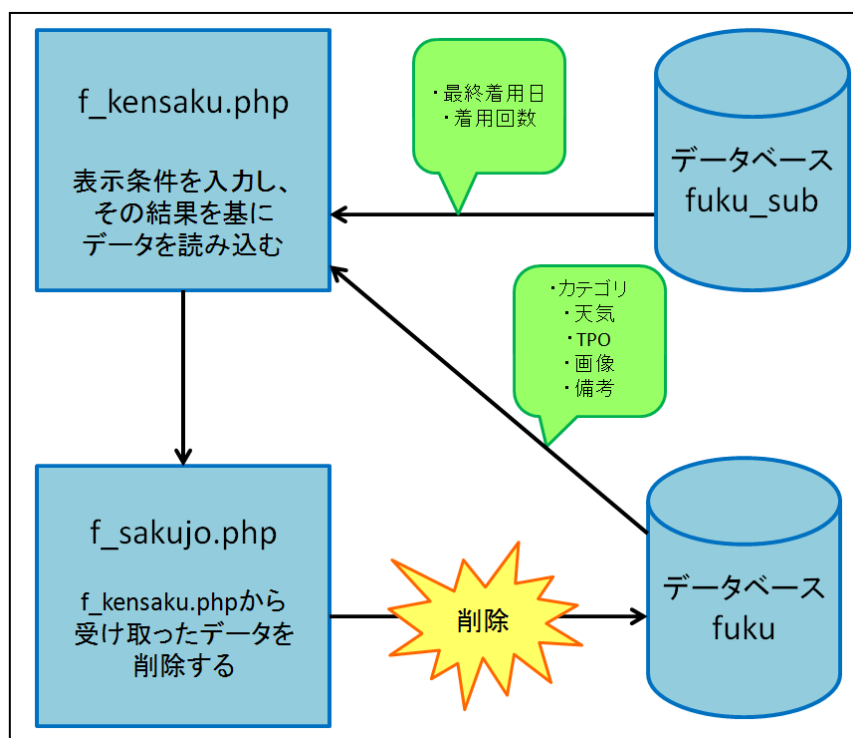


図 4-3-3-1 : アイテム検索・削除のデータの流れ

表 4-3-3-2 : f\_kensaku.php

```
<?php
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```

<html lang="ja">
<head>
    <title>アイテム検索</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>

<body>

<?php
extract($_POST);

echo "
<p><font size=¥"5¥">*アイテムを検索*</font></p>
<p>※必ず1以上チェックを付けてください。</p>
    <form method=¥"post¥" >
    <p>カテゴリ : <input type=¥"checkbox¥" name=¥"cate[]"¥" value=¥"1¥">トッ
プス
        <input type=¥"checkbox¥" name=¥"cate[]"¥" value=¥"2¥">ボトムス
        <input type=¥"checkbox¥" name=¥"cate[]"¥" value=¥"3¥">ワンピース
        <input type=¥"checkbox¥" name=¥"cate[]"¥" value=¥"4¥">アウター
        <input type=¥"checkbox¥" name=¥"cate[]"¥" value=¥"5¥">靴
        <input type=¥"checkbox¥" name=¥"cate[]"¥" value=¥"6¥">小物</p>

    <p>天気 : <input type=¥"checkbox¥" name=¥"ten[]"¥" value=¥"1¥">晴れ
        <input type=¥"checkbox¥" name=¥"ten[]"¥" value=¥"2¥">雨
        <input type=¥"checkbox¥" name=¥"ten[]"¥" value=¥"3¥">曇り
        <input type=¥"checkbox¥" name=¥"ten[]"¥" value=¥"4¥">雪
        <input type=¥"checkbox¥" name=¥"ten[]"¥" value=¥"5¥">指定なし
</p>

```



```
<p>TPO : <input type="checkbox" name="tpo[]" value="1">仕事
      <input type="checkbox" name="tpo[]" value="2">正装
      <input type="checkbox" name="tpo[]" value="3">運動
      <input type="checkbox" name="tpo[]" value="4">デート
      <input type="checkbox" name="tpo[]" value="5">その他
      <input type="checkbox" name="tpo[]" value="6">指定なし
</p>
```

```
<p>アイテム除外期間 : <select name="jogai">
      <option value="1">指定なし</option>
      <option value="2">3日</option>
      <option value="3">1週間</option>
      <option value="4">2週間</option></select></p>
```

```
<p>表示順 : <select name="hyouji">
      <option value="1">着用回数の少ない順</option>
      <option value="2">着用回数の多い順</option>
      <option value="3">着用日の昇順</option>
      <option value="4">着用日の降順</option>
      <option value="5">購入日の昇順</option>
      <option value="6">購入日の降順
</option></select></p>
```

```
<p><input type="hidden" name="num" value="1"></p>
<p><input type="submit" value="検索">
<input type="reset" value="リセット"></p><hr>
</form>;
```

```
mysql_connect('localhost', 'b04****', '*****');
```

```

mysql_select_db(' b04****' );
extract($_POST);

$where = '';
$tmp = array();
foreach((array)$_POST['cate'] as $ca) $tmp[] = ' fuku.cate="'. $ca. '";
if (!empty($tmp)) $where .= ' (.implode(' OR ', $tmp) . ')';

$tmp = array();
foreach((array)$_POST['ten'] as $te) $tmp[] = ' fuku.weather="'. $te. '";
if (!empty($tmp)) {if (!empty($where)) $where .= ' OR ';
$where .= ' (.implode(' OR ', $tmp) . ')';

$tmp = array();
foreach((array)$_POST['tpo'] as $tp) $tmp[] = ' fuku.tpo="'. $tp. '";
if (!empty($tmp)) {if (!empty($where)) $where .= ' OR ';
$where .= ' (.implode(' OR ', $tmp) . ')';
}}

if ($jogai == 2) {$where2 = "and saigo <= date_add(date(now()), interval -3 day)";}
elseif ($jogai == 3) {$where2 = "and saigo <= date_add(date(now()), interval
-7 day)";}
elseif ($jogai == 4) {$where2 = "and saigo <= date_add(date(now()), interval
-14 day)";}
else "";

if ($hyouji == 1) {$where3 = "order by kaisu asc";}
elseif ($hyouji == 2) {$where3 = "order by kaisu desc";}
elseif ($hyouji == 3) {$where3 = "order by saigo asc";}
elseif ($hyouji == 4) {$where3 = "order by saigo desc";}

```

```

elseif ($youji == 5) {$where3 = "order by kounyu asc";}
else {$where3 = "order by kounyu desc";}

if($num <> "1"){
    $sql="select
fuku. id, fuku. kounyu, fuku. cate, fuku. weather, fuku. tpo, fuku. pic, fuku. biko, fuku_
sub. kaisu, fuku_sub. saigo from fuku inner join fuku_sub on fuku. id = fuku_sub. id
where $where $where2 $where3";

    $result = mysql_query($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)) {
            echo "<p>";
            echo $row["id"];
            echo "<br />";
            echo "<img width=240 src=¥"pic/" . $row["pic"]. "¥">";
            echo "<br />";
            echo "購入日 : "          ;
            echo $row["kounyu"];
            echo "<br />";
            echo "カテゴリ : ";
                if ($row["cate"]==1) {echo "トッパス";}
                    elseif ($row["cate"]==2) {echo "ボトムス";}
                        elseif ($row["cate"]==3) {echo "ワンピース";}
                            elseif ($row["cate"]==4) {echo "アウター";}

```

```

        elseif ($row["cate"]==5) {echo "靴";}
        else {echo "小物";}

echo "<br />";
echo "天気 : ";

        if ($row["weather"]==1) {echo "晴れ";}
        elseif ($row["weather"]==2) {echo "雨";}
        elseif ($row["weather"]==3) {echo "曇り";}
        elseif ($row["weather"]==4) {echo "雪";}
        else {echo "指定なし";}

echo "<br />";
echo "TPO : ";

        if ($row["tpo"]==1) {echo "仕事";}
        elseif ($row["tpo"]==2) {echo "正装";}
        elseif ($row["tpo"]==3) {echo "運動";}
        elseif ($row["tpo"]==4) {echo "デート";}
        elseif ($row["tpo"]==5) {echo "その他";}
        else {echo "指定なし";}

echo "<br />";
echo "着用回数 : ";
echo $row["kaisu"];
echo "<br />";
echo "最終着用日 : ";

        if ($row["kaisu"]==0) {echo "未着用";}
        else {echo $row["saigo"]}

echo "<br />";
echo "備考 : ";
echo $row["biko"];
echo "</p>";
echo "<a href=¥"f_sakusei.php?id="";
echo $row["id"];

```

```

        echo "¥" . "> *コーディネート作成* </a>";
        echo "</p>";
        echo "<a href=¥" . f_sakujo.php?id=";
        echo $row["id"];
        echo "¥" . "> *この服を削除* </a><hr>";
    }
}
} else { echo "選択なし"; }
?>
</body>
</html>

```

表 4-3-3-3 : f\_sakujo.php

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="ja">
<head>
    <title>アイテム削除</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<?php
extract($_POST);
extract($_GET);
mysql_connect('localhost', 'b04****', '*****');

```

```

mysql_select_db('b04****');

if ($kakunin == "確認"){
    $sql = "delete from fuku where id = $id";
    mysql_query($sql);
    echo "アイテム削除完了！";
    echo "<p><a href=f_kensaku.php>検索に戻る</a></p>";
    exit;
}

$sql= "select * from fuku where id = $id";
$result = mysql_query($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)){
        echo "<p>このレコードを削除していいですか？</p>";
        echo '<form action = "" method="post">';

                echo "<p>";
                echo $row["id"];
                echo "<br />";
                echo "<img width=240 src=¥"pic/" . $row["pic"]. "¥">";
                echo "<br />";
                echo "購入日： "          ;
                echo $row["kounyu"];
                echo "<br />";
    }
}

```

```

echo "カテゴリ : ";

    if ($row["cate"]==1){echo "トップス";}

        elseif ($row["cate"]==2){echo "ボトムス";}

            elseif ($row["cate"]==3){echo "ワンピース";}

                elseif ($row["cate"]==4){echo "アウター";}

                    elseif ($row["cate"]==5){echo "靴";}

                        else {echo "小物";}

echo "<br />";

echo "天気 : ";

    if ($row["weather"]==1){echo "晴れ";}

        elseif ($row["weather"]==2){echo "雨";}

            elseif ($row["weather"]==3){echo "曇り";}

                elseif ($row["weather"]==4){echo "雪";}

                    else {echo "指定なし";}

echo "<br />";

echo "TPO : ";

    if ($row["tpo"]==1){echo "仕事";}

        elseif ($row["tpo"]==2){echo "正装";}

            elseif ($row["tpo"]==3){echo "運動";}

                elseif ($row["tpo"]==4){echo "デート";}

                    elseif ($row["tpo"]==5){echo "その他";}

                        else {echo "指定なし";}

echo "<br />";

echo "備考 : ";

echo $row["biko"];

echo "</p>";

echo '<input type = "hidden" name = "ren" value = ""';

echo $row["id"];

echo '"">';

echo '<p><input type="submit" name="kakunin" value="確認">';

```

```
    echo ' <input type="reset" value="リセット"></p>';  
    echo "</form>";  
  }  
}  
>  
  
</body>  
</html>
```

#### 4-3-4 コーディネート作成

アイテム検索で検索した服からコーディネートを作成する為のデータの流れを図 4-3-4-1、プログラムを表 4-3-4-2 に示す。

アイテムの全てのカテゴリ名と「未選択」という文字が表示されている。最初にアイテム検索で検索した服のカテゴリのみに「選択済」と表示されている。未選択をクリックすると再び服を検索できる。再び服の選択とする為のプログラムは表 4-3-4-3 に示す。服の検索、選択をコーディネートが完成するまで繰り返してコーディネートを作成していく。選択中の服は画像が表示されるのでコーディネート全体のイメージをしながらコーディネートの作成ができる。



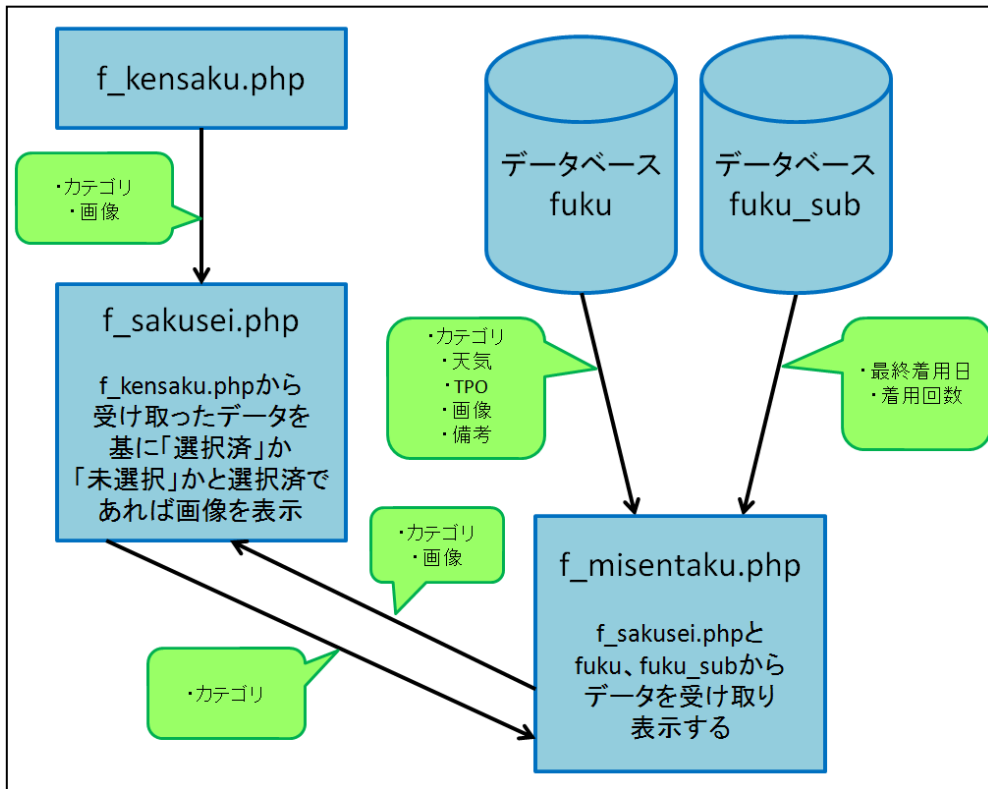


図 4-3-4-1 : コーディネート作成のデータの流れ

表 4-3-4-2 : f\_sakusei.php

```

<?php
session_start();
    ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
    <title>コーディネート作成</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>

<body>

<?php

```

```

echo "<p><font size=¥"5¥">* コーディネート作成 *</font></p>";
extract($_GET);
extract($_POST);
mysql_connect('localhost', 'b04****', '*****');
mysql_select_db('b04****');
$id = $_GET["id"];

$sql= "select * from fuku where id = $id";
$result = mysql_query($sql);
$rows = mysql_num_rows($result);

if($rows == 0) {
    echo "";
} else {
    while($row = mysql_fetch_array($result)) {
        if ($row["cate"] == 1) {
            $_SESSION["cate1"] = $id;
        }
        elseif ($row["cate"] == 2) {
            $_SESSION["cate2"] = $id;
        }
        elseif ($row["cate"] == 3) {
            $_SESSION["cate3"] = $id;
        }
        elseif ($row["cate"] == 4) {
            $Outer = 1;
            $_SESSION["cate4"] = $id;
        }
        elseif ($row["cate"] == 5) {

```

```

        $_SESSION["cate5"] = $id;
    }
    else {
        $_SESSION["cate6"] = $id;
    }
}

$cate1 = $_SESSION["cate1"];
$cate2 = $_SESSION["cate2"];
$cate3 = $_SESSION["cate3"];
$cate4 = $_SESSION["cate4"];
$cate5 = $_SESSION["cate5"];
$cate6 = $_SESSION["cate6"];

if (isset($cate1)) {echo "トップス : <font color='red'>選択済</font><br />";}
    else {
        echo "トップス : <a href=f_misentaku.php?cate=1>未選択</a><br />";
    }

if (isset($cate2)) {echo "ボトムス : <font color='red'>選択済</font><br />";}
    else {
        echo "ボトムス : <a href=f_misentaku.php?cate=2>未選択</a><br />";
    }

if (isset($cate3)) {echo "ワンピース : <font color='red'>選択済</font><br />";}
    else {
        echo "ワンピース : <a href=f_misentaku.php?cate=3>未選択</a><br />";
    }

if (isset($cate4)) {echo "アウター : <font color='red'>選択済</font><br />";}
    else {
        echo "アウター : <a href=f_misentaku.php?cate=4>未選択</a><br />";
    }

```

```

    }
if (isset($cate5)) {echo "靴 : <font color='red'>選択済</font><br />";}
    else {
        echo "靴 : <a href=f_misentaku.php?cate=5>未選択</a><br />";
    }
if (isset($cate6)) {echo "小物 : <font color='red'>選択済</font><br />";}
    else {
        echo "小物 : <a href=f_misentaku.php?cate=6>未選択</a><br />";
    }

echo ' <p><form method="post" action="f_coordinate.php" name="コーディネート完成" >
    <input type="hidden" name="fin" value="1">
    <input type="submit" value="コーディネート完成">
</form></p><hr />';

if(isset($catel)){
    $sql="select * from fuku where id = $catel";
    $result = mysql_query($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)){
            echo "<img width=240 src=¥"pic/" . $row["pic"]. "¥">";
        }
    }
}

```

```

if(isset($cate2)) {
    $sql="select * from fuku where id = $cate2";
    $result = mysql_query($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)) {
            echo "<img width=240 src=¥"pic/" . $row["pic"]. "¥">";
        }
    }
}

if(isset($cate3)) {
    $sql="select * from fuku where id = $cate3";
    $result = mysql_query($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)) {
            echo "<img width=240 src=¥"pic/" . $row["pic"]. "¥">";
        }
    }
}

if(isset($cate4)) {
    $sql="select * from fuku where id = $cate4";

```

```

$result = mysql_query($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)) {
            echo "<img width=240 src=¥"pic/" . $row["pic"]. "¥">";
        }
    }
}

if(isset($cate5)) {
    $sql="select * from fuku where id = $cate5";
    $result = mysql_query($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)) {
                echo "<img width=240 src=¥"pic/" . $row["pic"]. "¥">";
            }
        }
    }

if(isset($cate6)) {
    $sql="select * from fuku where id = $cate6";
    $result = mysql_query($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)) {
            echo "<img width=240 src=¥"pic/" . $row["pic"]. "¥">";
        }
    }
}
?>

</body>
</html>

```

表 4-3-4-3 : f\_misentaku.php

```

<?php
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
    <title>コーディネート</title>
</head>

<body>

<?php
echo "
<p><font size=¥"5¥">*アイテムを選択して下さい*</font></p>

```

```
<form method="post" >
```

```
<p>天気 : <input type="checkbox" name="ten[]" value="1">晴れ  
      <input type="checkbox" name="ten[]" value="2">雨  
      <input type="checkbox" name="ten[]" value="3">曇り  
      <input type="checkbox" name="ten[]" value="4">雪  
      <input type="checkbox" name="ten[]" value="5">指定なし
```

```
</p>
```

```
<p>TPO : <input type="checkbox" name="tpo[]" value="1">仕事  
      <input type="checkbox" name="tpo[]" value="2">正装  
      <input type="checkbox" name="tpo[]" value="3">運動  
      <input type="checkbox" name="tpo[]" value="4">デート  
      <input type="checkbox" name="tpo[]" value="5">その他  
      <input type="checkbox" name="tpo[]" value="6">指定なし
```

```
</p>
```

```
<p>アイテム除外期間 : <select name="jogai">  
      <option value="1">指定なし</option>  
      <option value="2">3日</option>  
      <option value="3">1週間</option>  
      <option value="4">2週間</option></select></p>
```

```
<p>表示順 : <select name="hyouji">  
      <option value="1">着用回数の少ない順</option>  
      <option value="2">着用回数の多い順</option>  
      <option value="3">着用日の昇順</option>  
      <option value="4">着用日の降順</option>  
      <option value="5">購入日の昇順</option>  
      <option value="6">購入日の降順
```



```
</option></select></p>
```

```
<p><input type="hidden" nam="num" value="1"></p>
```

```
<p><input type="submit" value="検索">
```

```
<input type="reset" value="リセット"></p><hr>
```

```
</form>";
```

```
mysql_connect('localhost', 'b04****', '*****');
```

```
mysql_select_db('b04****');
```

```
extract($_POST);
```

```
extract($_GET);
```

```
$cate = $_GET["cate"];
```

```
$where = '';
```

```
$tmp = array();
```

```
foreach((array)$_POST['ten'] as $te) $tmp[] = 'fuku.weather="'.$te.'";
```

```
if (!empty($tmp)) {if (!empty($where)) $where .= ' OR ';
```

```
$where .= '('.implode(' OR ', $tmp) . ')';
```

```
$tmp = array();
```

```
foreach((array)$_POST['tpo'] as $tp) $tmp[] = 'fuku.tpo="'.$tp.'";
```

```
if (!empty($tmp)) {if (!empty($where)) $where .= ' OR ';
```

```
$where .= '('.implode(' OR ', $tmp) . ')';
```

```
}}
```

```
if ($jogai == 2) {$where2 = "and saigo <= date_add(date(now()), interval -3 day)";}
```

```
elseif ($jogai == 3) {$where2 = "and saigo <= date_add(date(now()), interval  
-7 day)";}
```

```
elseif ($jogai == 4) {$where2 = "and saigo <= date_add(date(now()), interval  
-14 day)";}
```

```

else "";

if ($youji == 1){$where3 = "order by kaisu asc";}
    elseif ($youji == 2){$where3 = "order by kaisu desc";}
    elseif ($youji == 3){$where3 = "order by saigo asc";}
    elseif ($youji == 4){$where3 = "order by saigo desc";}
    elseif ($youji == 5){$where3 = "order by kounyu asc";}
    else {$where3 = "order by kounyu desc";}

if($num <> "1"){
    $sql="select
fuku. id, fuku. kounyu, fuku. cate, fuku. weather, fuku. tpo, fuku. pic, fuku. biko, fuku_
sub. kaisu, fuku_sub. saigo from fuku inner join fuku_sub on fuku. id = fuku_sub. id
where cate = $cate $where $where2 $where3";
    $result = mysql_query($sql);
    $rows = mysql_num_rows($result);

    if($rows == 0){
        echo "<p>該当データがありません。 </p>";
        echo $sql;
    }
    else {
        while($row = mysql_fetch_array($result)){
            echo "<p>";
            echo $row["id"];
            echo "<br />";
            echo "<img width=240 src=¥"pic/" . $row["pic"]. "¥">";
            echo "<br />";
            echo "購入日 : "      ;
        }
    }
}

```

```

echo $row["kounyu"];
echo "<br />";
echo "カテゴリ : ";
    if ($row["cate"]==1) {echo "トップス";}
        elseif ($row["cate"]==2) {echo "ボトムス";}
            elseif ($row["cate"]==3) {echo "ワンピース";}
                elseif ($row["cate"]==4) {echo "アウター";}
                    elseif ($row["cate"]==5) {echo "靴";}
                        else {echo "小物";}
echo "<br />";
echo "天気 : ";
    if ($row["weather"]==1) {echo "晴れ";}
        elseif ($row["weather"]==2) {echo "雨";}
            elseif ($row["weather"]==3) {echo "曇り";}
                elseif ($row["weather"]==4) {echo "雪";}
                    else {echo "指定なし";}
echo "<br />";
echo "TPO : ";
    if ($row["tpo"]==1) {echo "仕事";}
        elseif ($row["tpo"]==2) {echo "正装";}
            elseif ($row["tpo"]==3) {echo "運動";}
                elseif ($row["tpo"]==4) {echo "デート";}
                    elseif ($row["tpo"]==5) {echo "その他";}
                        else {echo "指定なし";}
echo "<br />";
echo "着用回数 : ";
echo $row["kaisu"];
echo "<br />";
echo "最終着用日 : ";
    if ($row["kaisu"]==0) {echo "未着用";}

```

```

                else {echo $row["saigo"];}

                echo "<br />";
                echo "備考：";
                echo $row["biko"];
                echo "</p>";
                echo "<a href=¥"f_sakusei.php?id=";
                echo $row["id"];
                echo "¥"> *コーディネート選択* </a>";
                echo "</p>";
                echo "<hr />";
            }
        }
    }else{echo "選択なし";}

?>

</body>
</html>

```

#### 4-3-5 コーディネート登録

作成したコーディネートを登録する為のデータの流れを図 4-3-5-1、プログラムを表 4-3-5-2 に示す。このプログラムは、データベース coordinate と連動してコーディネートの作成で「コーディネート完成」をクリックするとこのプログラムを実行させる。

(表 4-3-4-2 参照)

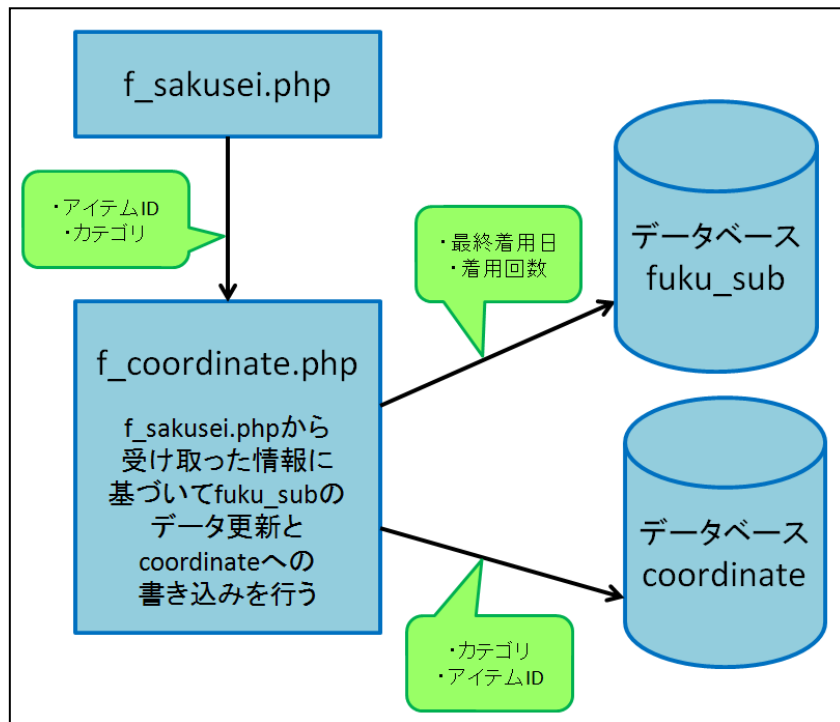


図 4-3-5-1 : コーディネート登録のデータの流れ

表 4-3-5-2 : `f_coordinate.php`

```

<?php
session_start();
    ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html lang="ja">
<head>
    <title>コーディネート登録</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>

<body>

<?php
extract($_POST);
  
```

```

extract($_GET);
mysql_connect('localhost','b04****','*****');
mysql_select_db('b04****');

$cate1= $_SESSION["cate1"];
$cate2= $_SESSION["cate2"];
$cate3= $_SESSION["cate3"];
$cate4= $_SESSION["cate4"];
$cate5= $_SESSION["cate5"];
$cate6= $_SESSION["cate6"];

$now = date("y-m-d");

if ($fin <> ''){
    $sql="insert into coordinate values(0, '$now', '$cate1', '$cate2',
'$cate3', '$cate4', '$cate5', '$cate6')";
    $sql1="update fuku_sub set kaisu = kaisu + 1,saigo = '$now' where id =
'$cate1' ";
    $sql2="update fuku_sub set kaisu = kaisu + 1,saigo = '$now' where id =
'$cate2' ";
    $sql3="update fuku_sub set kaisu = kaisu + 1,saigo = '$now' where id =
'$cate3' ";
    $sql4="update fuku_sub set kaisu = kaisu + 1,saigo = '$now' where id =
'$cate4' ";
    $sql5="update fuku_sub set kaisu = kaisu + 1,saigo = '$now' where id =
'$cate5' ";
    $sql6="update fuku_sub set kaisu = kaisu + 1,saigo = '$now' where id =
'$cate6' ";
    mysql_query($sql);
    mysql_query($sql1);

```

```
mysql_query($sql2);
mysql_query($sql3);
mysql_query($sql4);
mysql_query($sql5);
mysql_query($sql6);
echo "登録完了しました！";
$_SESSION = array();
session_destroy();
}

echo "<p><a href=menu2.html>コーディネートページに戻る</a></p>";
echo "<p><a href=top.html>トップページに戻る</a></p>";

?>

</body>
</html>
```

## 5. 評価と考察

本章では、第4章で実装したシステムが正常に動作するかの確認と考察をする。

### 5-1 動作確認

#### 5-1-1 トップページ

トップページ(top.html)を開いた画面が図 5-1-1-1 である。アイテム、コーディネートをクリックするとアイテムページ(menu1.html)とコーディネートページ(menu2.html)の各ページへ移動する。アイテムページとコーディネートページを次の図 5-1-1-2 と図 5-1-1-3 に示す。

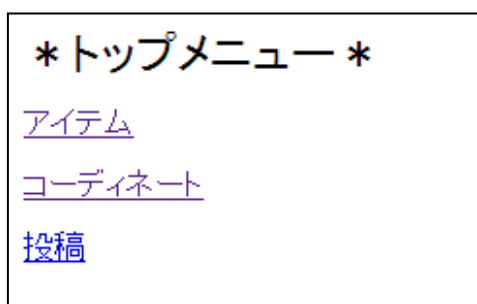


図 5-1-1-1 : トップメニュー画面

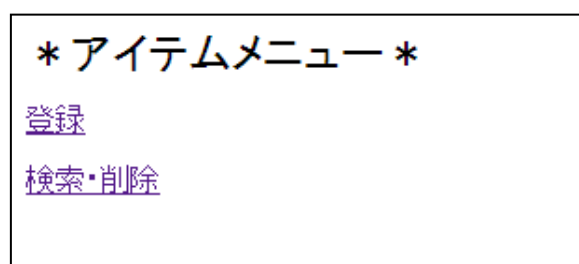


図 5-1-1-2 : アイテムページ画面

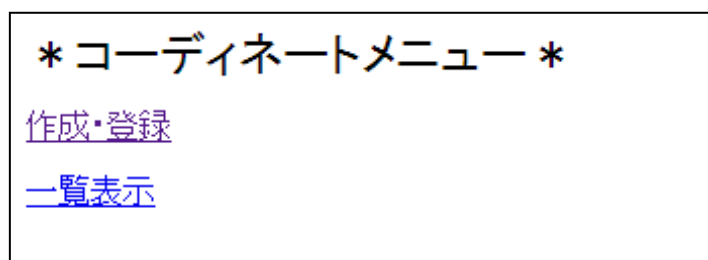


図 5-1-1-3 : コーディネートページ画面

#### 5-1-2 アイテム登録ページ

アイテムを登録する為の画面である。アイテムページで「登録」をクリックすると移動するアイテム登録画面を図 5-1-2-1 に示す(図 5-1-1-2 参照)。この画面でカテゴ



り、天気、TPO を選択している画面が図 5-1-2-2 である。また、アイテムの画像ファイルを選択している画面が図 5-1-2-3 である。さらに「登録」ボタンをクリックし、アイテム登録が完了した画面を図 5-1-2-4 に示す。

**\* データを入力 \***

購入日:  -  -

カテゴリ:

天気:

TPO:

画像:  選択されていません

備考:

図 5-1-2-1 : アイテム登録ページ画面

カテゴリ:

天気:

TPO:

画像:  選択されていません

図 5-1-2-2 : カテゴリ、天気、TPO の選択画面

開く

「ダウンロード」服写真 検索

整理 表示 折りたたむ 展開 詳細 設定

名前	撮影日	タグ	サイズ
NCM_0553.JPG			NCM_0584.JPG
NCM_0554.JPG			NCM_0585.JPG
NCM_0555.JPG			NCM_0586.JPG
NCM_0557.JPG			NCM_0587.JPG
NCM_0558.JPG			NCM_0588.JPG
NCM_0559.JPG			NCM_0589.JPG
NCM_0560.JPG			
NCM_0561.JPG			
NCM_0562.JPG			
NCM_0582.JPG			
NCM_0583.JPG			

ファイル名(N): NCM\_0553.JPG 全てのファイル

図 5-1-2-3 : アイテムの画像ファイルの選択画面

アップロード成功！正常に登録されました！  
[登録ページに戻る](#)

図 5-1-2-4：アイテム登録完了画面

### 5-1-3 アイテム検索ページ

アイテムを検索する為のページで、アイテムページ画面で「検索・削除」をクリックするとアイテム検索ページへ移動する(図 5-1-1-2 参照)。このアイテム検索ページの画面を図 5-1-3-1 に示す。このページではカテゴリ、天気、TPO をチェックボックスで選択できる。また、近日中に着用したアイテムを検索結果画面から除外するアイテム除外期間と検索結果の表示順はプルダウンメニューで選択できる。チェックボックスとプルダウンメニューで選択している画面をそれぞれ図 5-1-3-2 と図 5-1-3-3 に示す。さらに「検索」ボタンをクリックすると検索結果が表示される。その検索結果の画面が図 5-1-3-4 である。

**\* アイテムを検索 \***  
※必ず1以上チェックを付けてください。

カテゴリ:  トップス  ボトムス  ワンピース  アウター  靴  小物

天気:  晴れ  雨  曇り  雪  指定なし

TPO:  仕事  正装  運動  デート  その他  指定なし

アイテム除外期間:

表示順:

図 5-1-3-1：アイテム検索ページ画面

**\* アイテムを検索 \***  
※必ず1以上チェックを付けてください。

カテゴリ:  トップス  ボトムス  ワンピース  アウター  靴  小物

天気:  晴れ  雨  曇り  雪  指定なし

TPO:  仕事  正装  運動  デート  その他  指定なし

図 5-1-3-2：カテゴリ、天気、TPO 選択画面



図 5-1-3-3 : アイテム除外期間、表示順の選択画面



図 5-1-3-4 : 検索結果の表示画面

#### 5-1-4 アイテム削除ページ

アイテムを削除する為のページで、アイテム検索ページで検索結果が表示されると「この服を削除」というボタンもアイテム情報と一緒に表示される(図 5-1-3-4 参照)。このボタンをクリックすると図 5-1-4-1 の削除確認ページへと移動する。削除確認ページで「確認」ボタンをクリックして削除が完了した画面を図 5-1-4-2 に示す。



図 5-1-4-1：アイテム削除確認画面

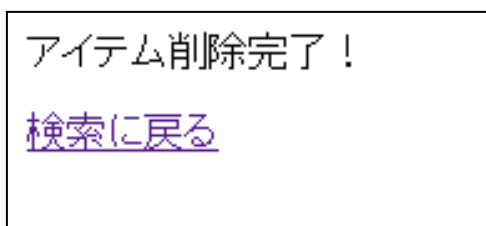


図 5-1-4-2：アイテム削除完了画面

#### 5-1-5 コーディネート作成・登録ページ

コーディネートページの「作成・登録」又は、アイテム検索ページの検索結果の「コーディネート作成」をクリックするとコーディネート作成ページへ移動する(図 5-1-1-3、図 5-1-3-4 参照)。コーディネートページから移動した場合の画面を図 5-1-5-1、アイテム検索ページから移動した場合の画面を図 5-1-5-2 に示す。コーディネート作成ページでは、アイテムのカテゴリ名と「選択済」か「未選択」の文字が表示されている。「選択済」のアイテムは下に画像が表示される。また、「未選択」の文字はクリックできる。「未選択」をクリックするとそのカテゴリのアイテム表示画面へ移動する。コーディネートを作成時の画面とみ「未選択」をクリックした後のアイテム表示画面をそれぞれ図 5-1-5-3 と図 5-1-5-4 に示す。また、「コーディネート完成」ボタンをクリックしてコーディネートを登録した画面が図 5-1-5-5 である。

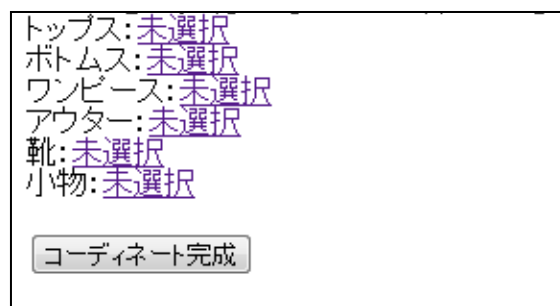


図 5-1-5-1 : コーディネートページからのコーディネート作成画面



図 5-1-5-1 : アイテム検索ページからのコーディネート作成画面



図 5-1-5-3 : コーディネート作成時画面

**\* アイテムを選択して下さい \***

天気:  晴れ  雨  曇り  雪  指定なし

TPO:  仕事  正装  運動  デート  その他  指定なし

アイテム除外期間:

表示順:

---

36



購入日: 2013-09-15  
 カテゴリ: 靴  
 天気: 指定なし  
 TPO: 指定なし  
 着用回数: 2  
 着用品ID: 0013\_10\_10

図 5-1-5-4 : 未選択のカテゴリのアイテムの選択時画面

**登録完了しました！**

[コーディネートページに戻る](#)

[トップページに戻る](#)

図 5-1-5-5 : コーディネート登録完了画面

## 5-2 機能評価

設計目標だった機能を評価したものを表 5-2 に示す。

表 5-2 : 実装結果一覧

番号	設計目標の機能	実装状況
1	アイテムの登録	○
2	アイテムの検索・表示	○
3	アイテムの削除	○
4	アイテムの提案	△

5	コーディネート作成・登録	○
6	コーディネートの表示	△
7	ソーシャル化	×

#### (1) アイテムの登録

購入日、カテゴリ、天気、TPO、画像、備考(メモすることがある時のみ)を入力し、登録する機能ができた。丸山さんの卒業研究と比較すると、丸山さんの研究のアイテムの登録機能は、登録時に入力する内容が多くて入力するのが面倒だった。この点を改善して、アイテムを提案する時に必要な最低限の情報のみを入力することができた。

#### (2) アイテムの検索・表示

アイテムをただ表示させるだけでなく、カテゴリ、天気、TPO をチェックボックスで絞って検索して表示させることができた。また、近日中に着た服を除外する機能を 3 日間、1 週間、2 週間と除外をしない指定なしの 4 つから選択させることができた。また、表示順を着用回数の少ない順、着用回数の多い順、着用日の昇順、着用日の降順、購入日の昇順、購入日の降順の 6 つから選択させることができた。アイテムを検索して絞ったり、除外期間や表示順を選択したりすることでコーディネートを作成する時に自分の好きな基準でアイテムを探すことができるのでアイテム提案に繋がる。また、アイテム情報を閲覧する時に閲覧したいアイテム情報をすぐに表示できる。欲しい情報だけに絞り、近日中に着用した服を除外し、検索結果の表示順を選択することでコーディネートの作成の時やアイテム情報の閲覧の時の時間短縮に繋がる。

#### (3) アイテム削除

間違ってアイテムの登録をしてしまったり、アイテムが必要なくなってしまう時の為にアイテムの削除機能を実装することができた。アイテムを削除するボタンをクリックしてすぐに削除ではなく、一度削除をしていいか確認してから削除するプログラムが完成した。

#### (4) アイテムの提案

アイテムの検索・表示でも述べた通り、アイテムを検索して絞ったり、除外期間や表示順を選択したりすることができたので、その日の天気や自分の気分で好きなよう

にアイテム絞って提案させることができた。しかし、ソーシャル化機能から他のユーザからの評価をコーディネート作成時の基準にしたり、過去のコーディネート表示からお気に入りのコーディネートをそのままその日のコーディネートにしたりするのが実現できなかった。

#### (5) コーディネートの作成・登録

コーディネートを作成し、登録することができた。画像表示で現在選択しているアイテムがわかったり、未選択のカテゴリのアイテムは未選択ボタンですぐにアイテム表示させたりして、わかりやすくコーディネートの作成・登録する機能ができた。誰でも簡単にいつでもどこでもコーディネートをできるようになった。

#### (6) コーディネートの表示

コーディネートの登録番号と着用日を表示することはできたが、画像を表示することができなかった。コーディネートに使用させているアイテムの画像が表示されないとどんなコーディネートかわからない。コーディネートは着回しが重要なのでコーディネートの画像を表示できれば、コーディネート作成時のアイテム選択の幅も広がる。

#### (7) ソーシャル化

自分のコーディネートを投稿し、他のユーザから評価をもらうことを設計目標としていたが、ソーシャル化ができなかった。ソーシャル化が実現できれば、他のユーザからの評価もコーディネート作成時の基準の1つに繋がる為、コーディネートの幅も広がるので、より一層コーディネートを楽しむことが可能である。

### 5-3 考察

本研究では、コーディネートの時間を短縮したり、煩わしさを軽減したりして毎日のコーディネートを楽しむ為のアイテム提案をすることを目的としている。目的を達成する為に目標としていた機能を全て実装することはできなかったが、アイテム登録、アイテム検索と表示、アイテム削除、コーディネート作成と登録の機能は実装することができた。

今回、コーディネートの画像表示とソーシャル化ができなかった為、本研究の目的



であるアイテム提案は完全に達成できなかった。コーディネート表示の時に着用日など他に画像が表示されているほうがコーディネートのイメージはしやすい。その為、過去のコーディネートをそのままその日のコーディネートにすることも可能になる。また、ソーシャル化が実現すれば、自分のコーディネートを他のユーザに評価してもらうことができるので、アイテム検索と表示の時に「評価の良い順」という並べ替えも可能になる。さらにソーシャル化では他のユーザのコーディネート閲覧できる為、自分の好みのコーディネートを真似してコーディネートを楽しむこともできる。このように今回実装できなかった機能が実現できれば、より一層たくさんのアイテムの提案が可能になり、コーディネートの幅が広がる。その為にこれらの機能を実装し、より良いアプリケーションを目指す必要がある。今回はコーディネートの時間を短縮と煩わしさを軽減の目標は達成されたが、実際に着用している写真を使ってコーディネートを作成したり、天気予報と連動したりなどの新たな機能を追加することで更なる時間の短縮と煩わしさを軽減は期待できる。

## 6. まとめと今後の課題

### 6-1 まとめ

本研究の目的は、毎日のコーディネートを楽しむ為のアイテム提案をすることでコーディネート作成時の時間短縮と煩わしさの軽減を目指した。コーディネートの現状を知る為に身近な人へのヒアリング調査や雑誌の着回し企画、既存のサービスについて調査と分析をして、問題点を明らかにした。そして、本研究の目的を達成して問題点を解決する為に必要な機能の検討をし、設計から実装まで行って評価した。

事前調査により、コーディネートの決定方法は人それぞれで同じ人でもその日の気分や環境などでコーディネートの決定基準は変化する為、コーディネートの決定には様々な方法が必要であることがわかった。また、同じ服を着用するのを避ける期間も人それぞれであることがわかった。これらのことから、コーディネートの決定方法と同じ服を避ける期間は選択できるようにしたほうが良いというのが問題点であった。この問題を解決するためにコーディネート作成する時に表示するアイテムはカテゴリ、天気、TPO から自由に選択できるようにし、同じ服を避ける期間と表示順を選択できるシステムを構築した。これにより、自分の好きな基準からアイテムを選択することが可能になった。

目的であるアイテム提案には、まだ足りない点があるが、コーディネート作成時の時間短縮と煩わしさの軽減に繋がる仕組みを作ることができた。

### 6-2 今後の課題

今回実装できなかったコーディネートの画像表示とソーシャル化の 2 つの機能の完成を目指す。

#### (1) コーディネートの画像表示

コーディネートを頭の中で想像するのより、実際にアイテムの画像を見るほうが早くコーディネートのイメージができる。よって、コーディネートの画像を表示させることでコーディネート作成時の時間の短縮と手間を省くことが可能になる。さらに、表示されている過去コーディネートの画像をそのままコーディネート登録をできるようになるとアイテム提案にも繋がる。

#### (3) ソーシャル化

コーディネートを投稿、閲覧、評価することを目指す。投稿した自分のコーディネ

ートを評価してもらうことでその評価をコーディネート作成時の基準にすることが可能である。また、他のユーザのコーディネートを閲覧することで自分の好みのコーディネートを真似したり、今まで自分になかったコーディネートのアイデアの発見ができたりする。また、他のユーザと交流することでファッション情報を入手することもできる。これらのことが可能になるとコーディネートを今まで以上に楽しめる。

以上を今後の課題として継続して改善に取り組んでいきたい。その結果、よりたくさんアイテムができるシステムを目指す。

参考文献

【1】 mini

宝島社 2013年3月号、2013年/8月号

【2】 sweet

宝島社 2013年/10月号

【3】 ViVi

講談社 2013年/9月号

【4】 おしゃれ天気

<https://play.google.com/store/apps/details?id=jp.co.recruit.mtl.osharetenki>

2013年7月23日

【5】 CodiSnap

<https://play.google.com/store/apps/details?id=jp.gmomeia.coordisnap>

2013年7月26日

【6】 FUKULOG

<http://fukulog.jp/>

2013年8月10日

【7】 suGATALOG

<http://www.ayakasato.com/sugatalog.html>

2013年9月15日

【8】 PHP マニュアル

<http://www.php.net/>

2013年11月29日

【9】 丸山みゆき「コーディネート支援のための所有衣服管理とアイテム提案機能の検討」

2012年度卒業研究

【10】 小島まさご「実践マスター PHP+MySQL」

ソーテック社 2007年

【11】 矢吹太郎、佐久田博司「Web アプリケーション構築入門(第2版) - 実践! Web ページ制作からマッシュアップまで」

森北出版 2011年

## 謝辞

本研究を進めるにあたり、ご指導を頂いた渡辺恭人准教授には大変感謝しております。論文の構成をはじめ、プログラミングや執筆に至るまでとても親身にご指導、ご指摘して頂きました。昨年度までのゼミナールではプログラミングの基礎からご指導頂き、本研究の実装で結果を出すことができました。今年度は論文の進め方や調査の方法など細部にわたり丁寧にご指導して下さいました。心より感謝致します。

最後に渡辺恭人准教授を始め、モニタリング調査に参加して頂いた方、渡辺ゼミナールの皆様など本研究に関わった全ての方に感謝を述べ、謝辞とさせていただきます。本当にありがとうございました。