

2013 年度卒業論文

「身近な人と食事情報の共有によるダイエット支援の検討」

指導教員：渡辺 恭人

1040010

茅根浩希

提出日：2012 年 12 月 16 日

|                                    |    |
|------------------------------------|----|
| 1. 背景と目的                           | 4  |
| 1-1 背景                             |    |
| 1-2 目的                             |    |
| 1-3 本論文の構成                         |    |
| 2. 現状と問題点                          | 5  |
| 2-1 現状                             |    |
| 2-2 太る原因と危険性                       |    |
| 2-3 既存のダイエット支援サービス                 |    |
| 2-3-1 スマートフォンアプリケーション              |    |
| 2-3-2 Web サービス                     |    |
| 2-4 既存のダイエット支援サービスを一定期間使用した比較と評価   |    |
| 2-4-1 既存のダイエット支援サービスの比較と評価         |    |
| 2-5 既存のダイエット支援サービスの問題点             |    |
| 3. 解決法の検討                          | 15 |
| 3-1 解決へのアプローチ                      |    |
| 3-2 摂取カロリーの記録を写真で簡単にする             |    |
| 3-3 体重の記録は1週間に1回だけ記録日を作る           |    |
| 3-4 グループ機能                         |    |
| 4. 設計と実装                           | 17 |
| 4-1 アプリケーションの設計                    |    |
| 4-1-1 アプリケーションの全体のイメージ             |    |
| 4-1-2 食品の写真を登録するイメージ               |    |
| 4-2 データベースの設計                      |    |
| 4-2-1 食べたものの管理テーブル                 |    |
| 4-2-2 アプリケーションへログインするためのアカウントのテーブル |    |
| 4-2-3 アカウントの情報のテーブル                |    |
| 4-2-4 体重を記録するテーブル                  |    |
| 4-3 アプリケーションの実装                    |    |
| 4-3-1 ログインページ                      |    |

|                     |    |
|---------------------|----|
| 4-3-2 アカウソトの登録      |    |
| 4-3-3 トツプページ        |    |
| 4-3-5 食べた食品の登録      |    |
| 4-3-6 登録された食品を見る    |    |
| 4-3-7 体重を登録         |    |
| 4-3-7 アカウソトの内容を変更する |    |
| 4-7-8 ログアウトする       |    |
| 5. 評価と考察            | 52 |
| 5-1 動作検証            |    |
| 5-2 評価              |    |
| 5-2-1 実装の評価         |    |
| 5-2-2 モニターの評価       |    |
| 5-3 考察              |    |
| 6. まとめと今後の課題        | 59 |
| 6-1 まとめ             |    |
| 6-2 今後の課題           |    |
| 6-2-1 グループ機能        |    |
| 6-2-2 近況の表示方法       |    |
| 6-2-3 データをグラフによる視覚化 |    |
| 参考文献                | 61 |
| 謝辞                  | 62 |

## 1章 背景と目的

### 1-1. 背景

今日、多くの人々が自分の体型や健康を気にしてさまざまな方法のダイエットに取り組んでいる。しかし、実際には気にしていても運動する機会が少なかったり、常に意識をしていなければ食べ過ぎたり、一人でダイエットをするのが辛くて失敗してしまう人も多い。筆者も、大学生になり自由な時間が多くなったが、サークルや部活動に所属していないため、運動する機会が少なくなった。それと反対に、友人との交流が多くなり食べる機会が増え、お互い体型が大きくなり体型の話をする機会が多くなった。さらに大学生から社会人になると、今よりも運動に時間をかける事が出来なくなり、さらに、運動する機会がなくなる。このような現状からダイエットをするためにまず、食生活を見直す事と一人ではなく仲間と取り組む必要がある。

### 1-2. 目的

本研究では、今の体の状況を把握するために食べ物や体重の記録・管理、家族や友人などの身の回りの人と記録を共有する事によって体重維持、ダイエットを長く続ける事の手助けにする事を目的とする。

### 1-3. 本論文の構成

本論文では、2章で、ダイエットの現状や種類、共通点を調べ、また既存のダイエット支援サービスの問題点の分析を行う。3章では、前章で明らかになった事を踏まえ、解決策の提案を行う。4章では、提案をもとにシステムの設計を行う。5章では、設計案をもとに実装を行う。6章では、実装や実際に使用した評価を行い、考察をする。7章では本研究をまとめ、今後の課題を述べる。以上が本論文の構成である。

## 2章 現状と問題点

ここでは、どのようなダイエットがどのように行われているのか現状を把握するため調査する。

### 2-1. 現状

ダイエットと言えば運動を連想する人は多くいる。実際にダイエットのためにジムに通う人や、スポーツをして汗を流す人たちが多くいる。しかし、体脂肪1kg減らすためには約7000キロカロリーから7200キロカロリーの消費が必要になる。体脂肪1kgを減らすためにランニングまたはウォーキングをする場合に摂取カロリーを抜いて考えると、約120kmもランニングまたはウォーキングをする必要があることが図2-1-1を見て分かる。このことから、運動だけによるダイエットの成功は難しいために、摂取するカロリーを調整する必要がある。

| エネルギー量       | 100kcal  | 160kcal                         | 250kcal                          |
|--------------|--|---------------------------------|----------------------------------|
| 食品例          | 缶コーヒー1本  | 大福もち 1個                         | ・アイスクリーム<br>コーンカップ1個<br>・ビール大瓶1本 |
| 運動 (60kg 男性) | 急ぎ歩き 30分<br>普通の歩行 45分<br>ゆっくりした歩行 45分<br>ラジオ体操 30分<br>自転車(普通の速さ) 40分 | ゆっくり水泳20分<br>サイクリング(時速10km) 50分 | エアロビックダンス 60分<br>ゴルフ(丘陵) 50分     |

図2-1-1:食品のエネルギー量と消費に要する運動量(東京都病院経済本部より引用)

また、現代のダイエットは、スマートフォンの普及によりスマートフォンのアプリケーションを利用し行うケースが多くなってきている。MACROMILL と InMobi が共同で全国のスマートフォンユーザの女性300人(10代以下85人、20代100人、30代75人、40代以上40人)を対象に調査したところ、図2-1-2のような結果になった。現在ダイエットをしている人が60%という事が分かった。また、180人のうち64%の人がスマートフォンのアプリケーションを使用している、または使用した事ある。スマートフォンのダイエット支援アプリケーションは主に食事制限やカロリーコントロールをするためのものが多く、手軽で簡単に記録をつける事が出来る事から利用する人が多くなってきている。

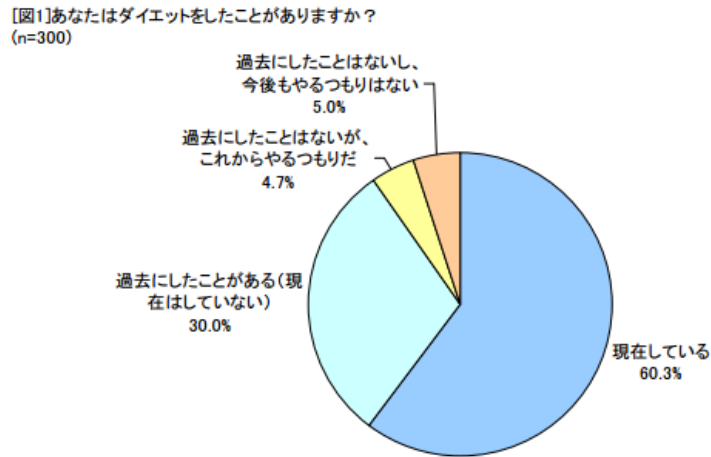


図 2-1-2: ダイエットの現状 (MACCROMILL×InMobi より引用)

## 2-2. 太る原因と危険性

太る原因として、食べ物を多量に食べてしまう事による、エネルギーの取りすぎと運動不足が肥満の主な原因だ。つまり、食事を多くとり摂取カロリーが多い、運動を不足により消費カロリーが少ない、またはその両方を満たす場合に肥満に陥りやすいという事だ。また、食事の回数や時間の不規則、夜にまとめて食事をするという事も肥満につながる。食事の回数や時間の不規則であると、体の中にいつ食べ物が入ってくるかわからないため、体がエネルギーを脂肪に溜め込んでしまうからである。夜は栄養の吸収率が高まるため、エネルギーを脂肪に溜め込みやすい。

国立循環器病研究センターの循環器病情報サービスでは、肥満の原因を次のように述べている。

「現代人の肥満は、食べすぎ、運動不足に加え、食べ方の異常、（摂食パターンの異常）、遺伝的体質、さらに食事や運動をした後や体温を一定に保つために、体は生理的に熱を発生しますが、この働きが低下している場合（熱産生障害）などが原因になっています。さらに、最近のライフスタイルの急激な変化が、大きく影響しているのはいうまでもありません。食べすぎると、当然の事ですが、摂取エネルギーが消費エネルギーを上回って、残りは貯蔵に回ってしまいます。イライラしたり、いつも不安感がある場合、それから逃避したいために、つい食べすぎてしまいがちです。運動が

不足すれば、消費エネルギーが減り、普通量の食事でも太る原因になります。肥満者の食事回数は、必ずしも多いというわけではなく、むしろ回数が少ない方が、太りやすい事もわかってきました。“至適体重”になるには朝食を抜いて、夜間にたくさん食べる“まとめ食い”や、1日の食事の半分以上を夜間にとる“夜食症候群”などが問題です。夜間は消化管の吸収機能が昼間より高まりやすく、エネルギーが貯蔵に回りやすいからです。」

(注1 <http://www.ncvc.go.jp/cvdinfo/pamphlet/obesity/pamph03.html#anchor-3>

2012年12月12日)

肥満には、糖尿病、循環器疾患や癌など多くの病気に関係していて、健康に重大な悪影響を及ぼす危険性がある。それを防ぐためには、生活または食事を正さなくてはいけない。

## 2-3. 既存のダイエット支援サービス

ここでは、既存のダイエット支援サービスについて調べる。

### 2-3-1. スマートフォンアプリケーション

人気があるスマートフォンのアプリケーションについて調べた。現状で述べたとおり、現代はスマートフォンのアプリケーションを使用しダイエットをしている人が多く、スマートフォンを持っていれば簡単に実行出来る手軽さが人気の理由だと考えられる。

#### (1). 32Calorie(作成者:h13i32mar 対応端末:Android 無料)

このアプリケーションは、最大摂取カロリーと目標摂取カロリーを自分の体重から算出し、朝食・昼食・夜食・間食の毎食の摂取カロリーをプラスボタンとマイナスボタンで操作し、自分が実際に摂ったカロリーを記録していくものだ。一か月の摂取カロリーを棒グラフとして表示する事ができ、平均も確認する事が出来る。また、ウィジェットも用意されおり、記録を確認する事がとても簡単だ。

使用手順は、図 2-3-1-1 の設定画面で1日に摂取するカロリーの範囲を自分で算出

し、目標に下限を入力し、最大に上限を入力する。



図 2-3-1-1:設定画面

カロリーの設定が終わると、図 2-3-1-2 の入力画面に切り替わり、自分で入力した目標と最大の目印が入ったグラフが表示される。実際に摂ったカロリーは朝食・昼食・夜食・間食ごとのプラスボタン・マイナスボタンを使用して操作する。食事ごとに色分けがされているので、摂取カロリーの内訳が一目でわかるようになっている。



図 2-3-1-2:入力画面

(2). メタボレコーダ (作成者:Next Future Inc 対応端末:Android 無料(有料版ある))

このアプリケーションは、体重を毎日記録していくものだ。しかし体重のほかにも、身長・基礎代謝・体脂肪量・筋肉量の入力が必要だ。これらを入力する事によって BMI・体脂肪率・筋肉率を知る事が出来る。このアプリケーションを使用するには、体重以



外にも体の状態を知る事が出来る体重計が必要になる。

記録したデータを 30 日間の測定データとしてグラフで表示する事や、Excel 形式でのデータ出力も可能だ。また PC 上でデータを登録する事も可能だ。このアプリケーションを使用するにはユーザ登録が必要になる。

アプリケーションを開くと図 2-3-1-3 のように入力画面が表示され、身長・基礎代謝・体脂肪量・筋肉量を入力する事が出来る。



図 2-3-1-3:入力画面

図 2-3-1-3 はメタボレコーダをスマートフォンで開いた場合のメニュー画面だ。図 2-3-1-5 の Web の入力画面と比べると機能は少ないがスマートフォンで今までのデータのグラフを見る事が出来る。



図 2-3-1-4:メニュー画面

Web 上でメタボレコーダを開くと、図 2-3-1-5 の入力画面が表示される。また 9 種類のボディタイプの中から毎日入力している体脂肪率と BMI の割合によって判定さ

れ自分がどのようなボディタイプなのか知る事が出来る。



図 2-3-1-5:Web の入力画面

## 2-3-2. Web サービス

Web サービスを提供している企業が多くあり、機能の充実から高い効果が得られる事が期待出来る。

### (1). GirlsDiet (作成者:株式会社 IBJ 無料)

GirlsDiet は株式会社 IBJ が運営する女性限定の SNS であり、日記を書くと同時に体重や体脂肪率を記録する事によって体重の変動を毎日記録する事が出来る。他の SNS と同じくコミュニティがあり、またダイエット成功者の座談会なども開かれる。

ダイエットのプランは自分で立てるが、成功者のダイエット法を参考にすることが多い。また、この SNS には設定した身長、体重や体脂肪率から自分の体型を推測しアバターに反映される。体重や体脂肪率が変わるとアバターの体形が変わるため、痩せた事や太った事が実感出来る。

図 2-3-2-1 では日記とデータ入力をする必要がある。日記には主に食べた物、実行した運動やエクササイズまたは自分が実行しているダイエットの方法などを中心に書き、配信する。



図 2-3-2-1:日記とデータ入力画面

他のユーザの日記を見る事によってその人が実行しているダイエットの種類ややり方を知る事が出来る。そして、ダイエット成功者の日記を参考に同じようにダイエットをする事が出来る。

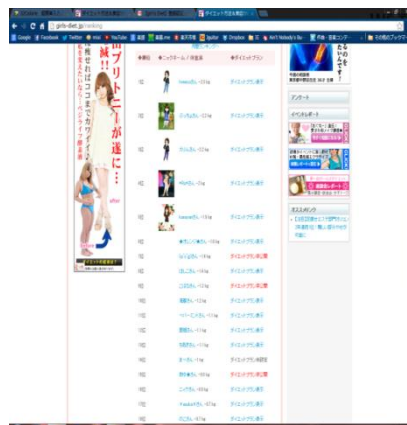


図 2-3-2-2:他のユーザの日記(観覧数ランキング)

## (2). あすけんダイエット(作成者:あすけん 無料)

株式会社ウィットが運営するサービスであり、栄養士のアドバイスが受けられダイエットをサポートしてもらえるところが特徴だ。主なサービスの流れは、食事、運動、体の状態を記録し栄養士からアドバイスが届き、アドバイスに従って食生活や運動習慣の改善をする事だ。他のサービスとの違いは、カテゴリーから分類を選択し、メニューを選び、分量を選ぶだけでカロリー・栄養素を自動計算されるところだ。

図 2-3-2-3 の入力画面では朝食・昼食・夕食・間食・運動の種類と量を入力が出来

る。



図 2-3-2-3:入力画面

図 2-3-2-3 で食事の中からボタンを押すと、図 2-3-2-4 の選択画面に移動する。そして、カテゴリーからとった食事の種類を選び、次に種類から食品名を選択して分量を選ぶ事で入力が完了する。



図 2-3-2-4:選択画面

## 2-4. 既存のダイエット支援サービスを一定期間使用した比較と評価

上で調べた既存のダイエット支援サービスを長期休みに一定期間使用し、そのサービスの比較と評価を行う。

### 2-4-1. 既存のダイエット支援サービスの比較と評価

32Calorie、メタボレコーダ、girlsDiet、あすけんダイエットの4つのダイエット支援サービスの機能を、比較すると図 2-4-1 のようになる。32Calorie とメタボレコー

ダのスマートフォンアプリケーションの機能は的が絞れており、一つの目的に一つのアプリケーションを使用しなければならない。それと反対に、Web サービスの場合は、記録や計算、コミュニティなどの幅広い目的を一つのサービスで実行する事ができた。

スマートフォンアプリケーションと Web サービスを一定期間使用した。スマートフォンは常に手元にあるため、どこにいてもアプリケーションを起動し素早く記録をする事ができ、ウィジェットを有効に活用する事により、アプリケーションを開かずに、データを確認する事が出来た。また、Web サービスの利点は、PC 上でもスマートフォンでも Web サービスを受ける事が出来る事だが、スマートフォンアプリケーションと比較すると記録を入力する事に時間がかかってしまった。

しかし、あすけんダイエット以外の 32Calorie、メタボレコーダや girlsDiet は自分で打ち込む項目が多くあり面倒と感じた。摂取カロリーも自分で調べ打ち込む事や、体重を毎日記録がする事によって、効果の表れを感じにくいために、挫折に繋がってしまう原因になる。また、あすけんダイエットでは、カテゴリー・メニュー・分量を一つずつ選んでいかなければいけないところから選択する数が多すぎて面倒と感じた。

|                    | 32Calorie | メタボレコーダ | girlsDiet | あすけんダイエット |
|--------------------|-----------|---------|-----------|-----------|
| 一日に必要な摂取カロリーの計算・記録 | △<br>記録だけ | ×       | △<br>記録だけ | ○         |
| 摂取したカロリーを記録する      | ○         | ×       | ○         | ○         |
| 体重管理               | ×         | ○       | ○         | ○         |
| BMI計算・記録           | ×         | ○       | ○         | ○         |
| コミュニティ             | ×         | ×       | ○         | ○         |

図 2-4-1:機能の比較

## 2-5. 既存のダイエット支援サービスの問題点

2-4 節の既存のダイエット支援サービスの評価を踏まえて、支援サービスの問題点は、摂取した食品のカロリーを調べ打ち込む事や、毎日体重の記録が繁雑であった。あす

けんダイエットでの食事の入力でもあったが、自分で細かく数字を打ち込む作業はないが食品を選択する事にとっても時間がかかった。そのために、時間をかけずに簡単に食品を登録出来る仕組みが必要である。

また、体重の入力を毎日行う必要はあるのだろうか。積み重ねは大事な事だが、体重はダイエットを始めたからといってすぐに落ちるものではない。毎日変わらない体重を打ち続ける事によって、飽きが生じダイエットを続ける妨げになってしまう。

girlsDiet やあすけんダイエットなどの SNS にはダイエットコミュニティが多くある。コミュニティには、同じダイエット方法を試している人の集団や、失敗をしないために励まし合うものを目的としたものがある。しかし、SNS のコミュニティに集まる同じ環境で同じ状況の面識のない人とダイエットをしても、慣れ合いになり、継続する事は難しい。

### 3章 解決法の検討

2-5 節の既存のダイエット支援サービスの問題点で取り上げた、食品のカロリーの入力・毎日の体重の入力・コミュニティの解決法を検討する。

#### 3-1. 解決へのアプローチ

2章で取り上げた既存の支援サービスの問題点は、入力に手間がかかるという事と、コミュニティは馴れ合いの場で継続にはつながらないという事だ。

入力に手間がかかってしまうのは、多くある選択肢から選ぶ事や、体重を毎日入力する事が原因であると考えられる。そこで私は、簡単に入力をする方法として、摂取した食品の写真を撮り記録しておく事と、体重の記録を毎日ではなく1週間に1回の記録日を作る事を解決策として提案する。

また、ダイエットをより長く継続出来るように小規模グループを作る事を提案する。

#### 3-2. 摂取カロリーの記録を写真で簡単にする

食べた食品のカロリーを調べて入力をするよりも、食べた食品を写真で撮る事によって、食べた食品が何なのか・どの量なのか・どのくらいカロリーがありそうなのかが見て分かり、文字だけで見るよりも食品の内容の記録を簡単に出来る。

また、食事毎に文字を打つよりも、今から食べる写真を撮る方が、より長く続けられる事に繋がる。

#### 3-3. 体重の記録は1週間に1回だけ記録日を作る

体重はすぐに変動しないが、それを毎日打ち続ける事によって、ダイエットへの飽きやモチベーションの低下が考えられるが、体重の記録を1週間に1回、記録日を作る事によってより高いモチベーションを維持出来る。

#### 3-4. グループ機能

本来ならば、グループはコミュニティに入るのだが本研究では、グループとコミュニティを分けて考える。本研究でのコミュニティとは顔見知りがいなく自由にコミュニティという場所に入出入り出来ると位置付け、それと反対にグループは顔見知りと同

意をもって入る事が出来る場所と位置付ける。

顔見知りの人とグループを作る理由は、自分の身の回りには自分の体型を知っていて、体の変化は見て確認する事が出来る。顔見知りの人達とグループを作る事によって、友人に食生活を見られるので規則正しく、また食べ過ぎに注意するように食事をする意識をさせる事が出来る。

|                    | 32Calorie | メタボレコーダ | girlsDiet | あすけんダイエット | 本研究で検討するアプリ |
|--------------------|-----------|---------|-----------|-----------|-------------|
| 一日に必要な摂取カロリーの計算・記録 | △<br>記録だけ | ×       | △<br>記録だけ | ○         | ○           |
| 摂取したカロリーを記録する      | ○         | ×       | ○         | ○         | ○           |
| グループ機能             | ×         | ×       | ×         | ×         | ○           |
| 体重管理               | ×         | ○       | ○         | ○         | ○           |
| BMI計算・記録           | ×         | ○       | ○         | ○         | ○           |
| コミュニティ             | ×         | ×       | ○         | ○         | ×           |

図 3-4:機能の比較



## 4章 設計と実装

2章の現状と問題点と3章での解決法を踏まえてアプリケーションの設計と実装を行う。

### 4-1. アプリケーションの設計

#### 4-1-1. アプリケーションの全体のイメージ

アプリケーションの全体のイメージは図4-1-1のようになる。最初にログイン画面を表示する。正しいIDとパスワードを入力し確認されるとユーザ画面に移る。IDとパスワードを持っていない場合は、新規会員登録画面へ移る。

ユーザ画面では、食品の登録・自分または友人の登録した食品を見る・体重の登録・ユーザの登録内容の変更が出来る。

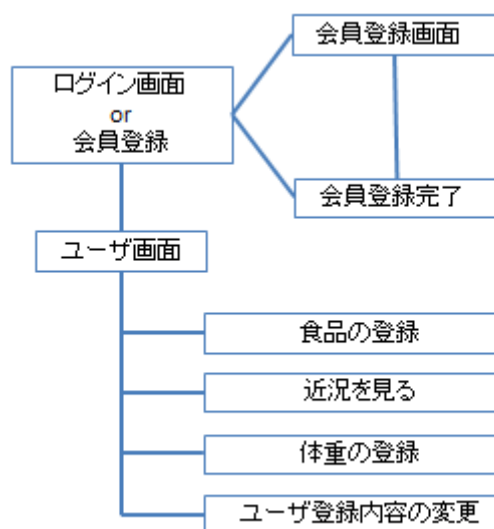


図 4-1-1:全体のイメージ

#### 4-1-2. 食品の写真を登録するイメージ

Aというユーザが写真を登録する場合のイメージは、入力フォームで写真を選択し、内容の送信を行う。内容を送信すると画像フォルダに画像を格納し、画像の名前をデータベースに格納する。そして内容を見る場合では、データベースから画像の名前を呼び出し、画像フォルダから画像を読み込むようにする。

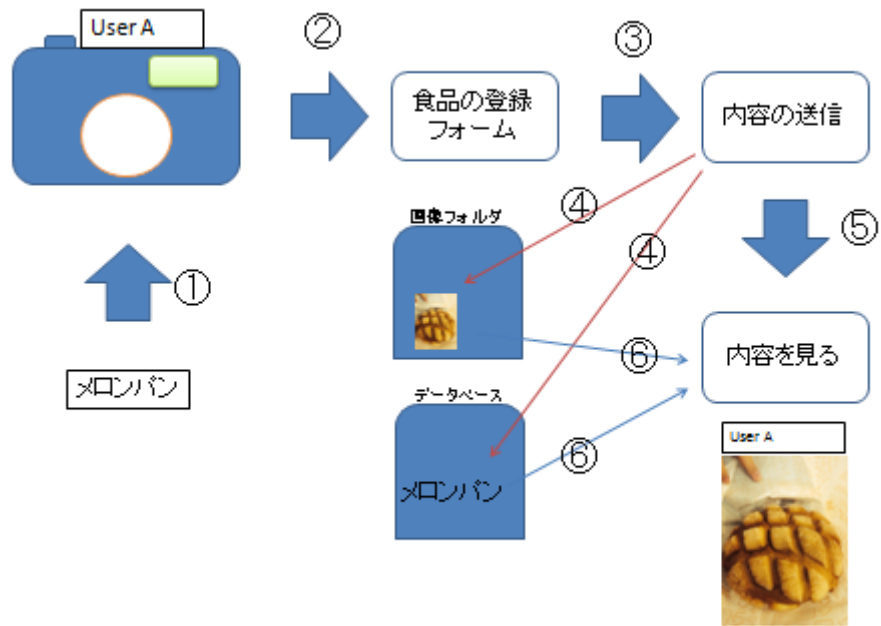


図 4-1-2:登録イメージ

#### 4-2. データベースの設計

必要なデータベースを設計し、作成する。

表 4-2:作成するテーブル名

| テーブル名    |
|----------|
| Ddb      |
| User     |
| Userdata |
| taiju    |

##### 4-2-1. 食べたものの管理テーブル

食品を登録するために必要なテーブルを設計する。

表 4-2-1-1:ddb テーブル設計

| 項目名    | フィールド名 | データ型         |
|--------|--------|--------------|
| ユーザ ID | uname  | varchar(255) |
| 食べた日   | day    | date         |

|         |        |              |
|---------|--------|--------------|
| 食べた時間   | tim    | time         |
| 食べ物名    | name   | varchar(255) |
| 食べた物の画像 | img    | varchar(255) |
| 推定カロリー  | kcal   | int(11)      |
| 報告      | hyou   | int(11)      |
| 連番      | renban | Int(11)      |

設計した表 4-2-1-1 をもとにテーブルを作成する。Renban には連番の設定をする。

表 4-2-1-2:ddb テーブル作成

```
mysql> create table ddb (uname varchar(255),day date,tim time,name
varchar(255)
, img varchar(255),kcal int(11),jtai int(11),renban int(11));
Query OK, 0 rows affected (0.01 sec)

mysql> alter table closet add index ind1(renban);
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table closet modify renban int auto_increment;
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show fields from ddb;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| uname | varchar(255) | YES  |     | NULL    |                |
| day   | date         | YES  |     | NULL    |                |
| tim   | time        | YES  |     | NULL    |                |
```

|                                       |              |     |     |      |                |
|---------------------------------------|--------------|-----|-----|------|----------------|
| name                                  | varchar(255) | YES |     | NULL |                |
| img                                   | varchar(255) | YES |     | NULL |                |
| kcal                                  | int(11)      | YES |     | NULL |                |
| jtai                                  | int(11)      | YES |     | NULL |                |
| renban                                | int(11)      | NO  | MUL | NULL | auto_increment |
| +-----+-----+-----+-----+-----+-----+ |              |     |     |      |                |

#### 4-2-2. アプリケーションへログインするためのアカウントのテーブル

アプリケーションへログインするためのユーザ ID やパスワードを保存しておくテーブルを設計する。

表 4-2-2-1:user テーブル設計

| 項目名    | フィールド名 | データ型      |
|--------|--------|-----------|
| 連番     | Id     | Int(11)   |
| ユーザ ID | Uname  | Char(255) |
| パスワード  | Pwd    | Char(255) |

表 4-2-2-1 をもとに、テーブルを作成する。Id には連番の設定をしておく。

表 4-2-2-2:user テーブル作成

```
mysql> create table user (id int,uname varchar(255),pwd varchar(255));
Query OK, 0 rows affected (0.00 sec)

mysql> alter table user add index ind1(id);
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table user modify id int auto_increment;
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show fields from user;
+-----+-----+-----+-----+-----+-----+
-----+
| Field | Type      | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
-----+
| id    | int(11)   | NO   | MUL | NULL     | auto_increment |
| uname | char(255) | YES  |     | NULL     |                |
| pwd   | char(255) | YES  |     | NULL     |                |
+-----+-----+-----+-----+-----+-----+
-----+
3 rows in set (0.00 sec)
```

4-2-3. アカウムの情報のテーブル

ユーザの情報を保存しておくテーブルの設計をする。

表 4-2-3-1:userdata テーブル設計

| 項目名       | フィールド名   | データ型          |
|-----------|----------|---------------|
| ユーザ ID    | Uname    | Varchar (255) |
| ユーザの名前    | Username | Varchar (255) |
| 目標体重      | Mokutai  | Int (11)      |
| 目標カロリーの下限 | Mokukc   | Int (11)      |
| 目標カロリーの上限 | Mokukc2  | Int (11)      |

表 4-2-3-1 をもとにテーブルを作成する。

表 4-2-3-2:userdata テーブル作成

```
mysql> create table userdata2 (uname varchar(255), username
varchar(255), mokutai int(11), mokukc int(11), mokukc2 int(11));
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> show fields from userdata;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| uname      | varchar(255)   | YES  |     | NULL    |       |
| username   | varchar(255)   | YES  |     | NULL    |       |
| mokutai    | int(11)        | YES  |     | NULL    |       |
| mokukc     | int(11)        | YES  |     | NULL    |       |
| mokukc2    | int(11)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

#### 4-2-4. 体重を記録するテーブル

ユーザの体重を管理するテーブルの設計をする。

表 4-2-4-1: taiju テーブル設計

| 項目名    | フィールド名 | データ型         |
|--------|--------|--------------|
| 登録日    | Tday   | Date         |
| ユーザ ID | Uname  | Varchar(255) |
| 現在の体重  | Taiju  | Int(11)      |

表 4-2-4-1 をもとにテーブルを作成する。

表 4-2-4-2: taiju テーブル作成

```
mysql> create table taiju (tday date,uname varchar(255),taiju int(11));
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> show fields from taiju;
```

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

```

|                                       |              |     |  |      |  |  |
|---------------------------------------|--------------|-----|--|------|--|--|
| tday                                  | date         | YES |  | NULL |  |  |
| uname                                 | varchar(255) | YES |  | NULL |  |  |
| taiju                                 | int(11)      | YES |  | NULL |  |  |
| +-----+-----+-----+-----+-----+-----+ |              |     |  |      |  |  |
| 3 rows in set (0.00 sec)              |              |     |  |      |  |  |

4-3. アプリケーションの実装

アプリケーションの設計をもとに実装を行う。

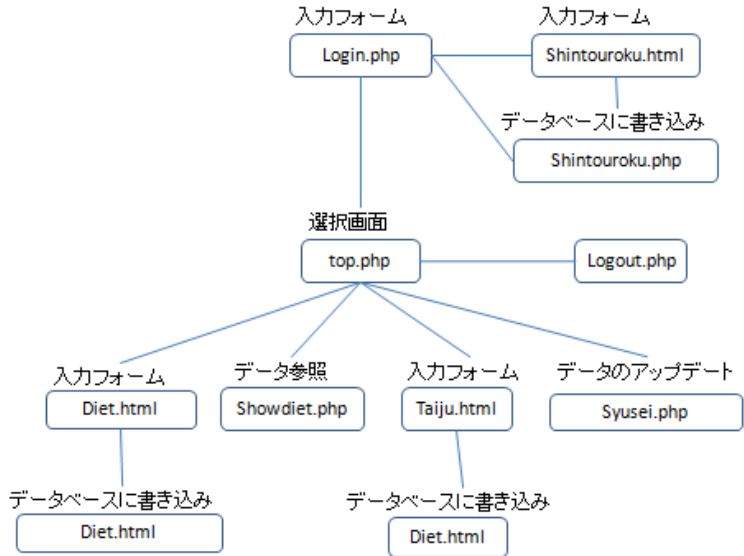


図 4-3: 実装のシステム構成

実装環境を表 4-3-1 にクライアントの環境、表 4-3-2 にサーバの環境、表 4-3-3 に使用したプログラム言語を示す。

表 4-3-1: クライアント環境

|              |   |
|--------------|---|
| オペレーティングシステム | Windows Vista™ Home Premium service pack 2    |
| CPU          | Intel® core™ 2 Duo CPU P8600 @2.40GHz 2.40GHz |
| メモリ          | 4.00GB  |

表 4-3-2: 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 1.6GHz |
| メモリ     | 1GB              |
| Web サーバ | Apache 2.2.8     |
| データベース  | MySql 5.0.96     |

表 4-3-3:プログラミング言語

|      |                 |
|------|-----------------|
| PHP  | Version:5.2.4-2 |
| HTML | HTML5           |

#### 4-3-1. ログインページ

ログインシステムを構築する。Login.php で、入力されたユーザ ID とパスワードを user のデータベースに問い合わせ、照合し一致した場合にトップ画面へ移るプログラムを作成する。セッションを使用し、ログインが成功した場合にユーザ ID を \$\_SESSION[ 'name' ]の中に保存し、ページを移動した場合にログイン状態のチェックを行う。また、次回から自動ログインするにチェックがされた状態でログインボタンを押すと、データが暗号化され cookie に保存される。

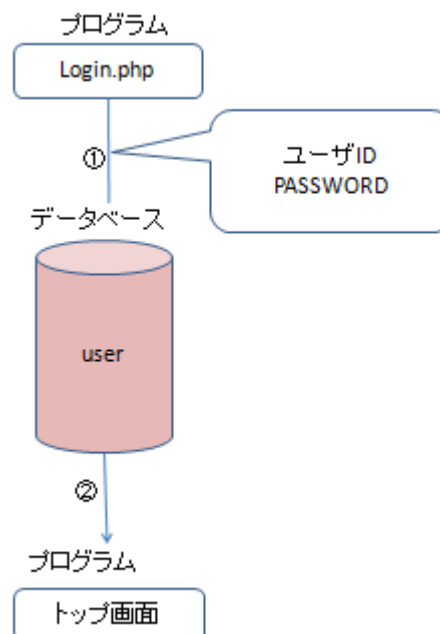


図 4-3-1-1:ログインのブロック図





図 4-3-1-2: ログイン画面

表 4-3-1-3: login.php

```
<?php
define("PASSWORD", "pass");
$message = "";
session_start();
extract($_POST);
mysql_connect('localhost', 'b040010', '*****');
mysql_select_db('b040010');
$sql = "select * from user where uname=' $id' and pwd=' $pwd' ";
$result = mysql_query($sql);

if(isset($_COOKIE["TEST_COOKIE"])    &&    $_COOKIE["TEST_COOKIE"]    ===
sha1(PASSWORD)) {

    $_SESSION["TEST"] = $_COOKIE["TEST_COOKIE"];
    header("Location:top.php");
}
```

```

if(isset($_POST["action"])&&$_POST["action"]=="login"){
    if(mysql_num_rows($result)!=0){//パスワード確認

        $_SESSION["TEST"] = md5(PASSWORD);
        //暗号化してセッションに保存

        if(isset($_POST["memo"]) && $_POST["memo"]=="true"){
            //次回からは自動的にログイン
            setcookie("TEST_COOKIE", $_SESSION["TEST"], time()+3600*24*14);
            //暗号化してクッキーに保存
            $_SESSION["uname"] = $_POST["id"];
        }
        $_SESSION["uname"] = $_POST["id"];
        header("Location:top.php");

    }else{
        session_destroy();//セッション破棄
        $message = "ID または PASSWORD が違います";
    }
}

?>
<!DOCTYPE HTML PUBLIC>
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8">

```

```

</head>
<body>
<h1>Login</h1>
<?php
    if($message!=""){
        print "<p class='message'>". $message. "</p>\n";
    }
?>
<form action="" method="post">
<p>ID : <input name="id" type="text" value="" /></p>
<p>PASSWORD : <input name="pwd" type="password" value="" /></p>
<p><label><input type="checkbox" name="memo" value="true" />次回からは自動的にログインする</label></p>
    <input name="action" type="submit" value="login" /></p>
    <a href = shintouroku.html>新規登録</a>
</form>
</body>
</html>

```

#### 4-3-2. アカウントの登録

ログインするためのユーザ ID とパスワードの登録を行うページを作成する。

shintouroku.html でユーザ ID ・パスワード ・氏名 ・目標体重のデータを入力し shintouroku.php に送り、ユーザ ID とパスワードは user のデータベースに、氏名と目標体重は userdata のデータベースに書き込む。また、目標体重に 25 と 30 を掛けた数を userdata の mokukc のフィールドと mokukc2 のフィールドへ保存する。

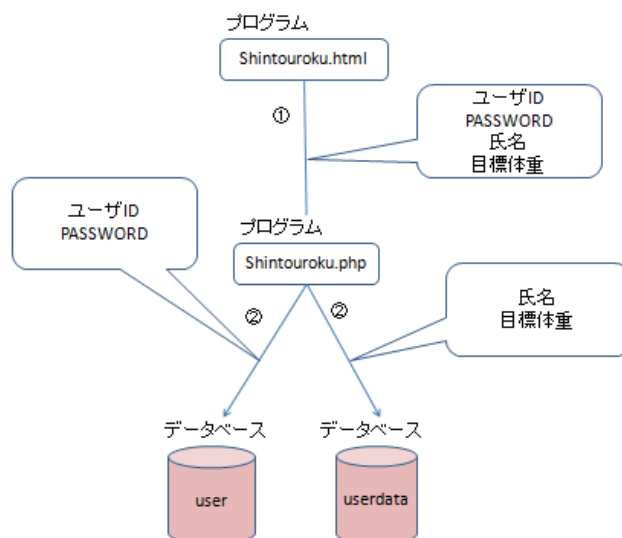


図 4-3-2-1:新規登録のブロック図



図 4-3-2-2:新規登録画面

表 4-3-2-3:Shintouroku.html

```

<!DOCTYPE HTML PUBLIC>
<html lang="ja">
<head>
  <title>ダイエット</title>
  <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8">
</head>

```

```

<body>
<h1>新規登録</h1>
<form name="shintouroku" action="shintouroku.php" method="post">
<p>ID : <input type="text" name="id" size="20" value="" /></p>
<p>PASSWORD : <input type="text" name="pwd" size="20" value="" /></p>
<p>名前 : <input type="text" name="nam" size="20" value="" /></p>
<p>目標体重 : <input type="text" name="moku" size="20" value="" /></p>
<p><input type="submit" value="登録"></p><br />
<p><a href="top.php">トップにもどる</a></p>

</form>
</body>
</html>

```

表 4-3-2-4:Shintouruku.php

```

<!DOCTYPE HTML PUBLIC>
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8">
</head>
<body>

<?php
extract($_POST);
mysql_connect('localhost', 'b040010', '*****');
mysql_select_db('b040010');

```

```

$username = $_POST["id"];
$password = $_POST["pwd"];
$cpwd = md5($password);
$mokuk = $_POST["moku"] * 25;
$mokuk2 = $_POST["moku"] * 30;
$sql = "insert into user values (0,'$username','$pwd')";
$sql2 = "insert into userdata values
('$username','$nam','$moku','$mokuk','$mokuk2')";
mysql_query($sql2);
$result = mysql_query($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 "<a href = shintouroku.html>戻る</a>";
    echo "</p>";
} else {
    echo "登録完了<br />";
    echo "<a href = login.php>ログイン画面に戻る</a>";
}
?>
</body>
</html>

```

#### 4-3-3. トップページ

トップページでは user と userdata のデータベースから新規会員ページで登録した名前と目標体重、目標体重から算出した 1 日の目安の摂取カロリーをトップページに表示する。また体重登録していた場合に、登録した体重から目標体重を引き目標まであ

とどのくらいなのかを表示させる。

トップページから食品の登録、登録された内容を見る、体重の登録、登録内容の変更のページに移動出来るようにする。なお、体重の登録ページは今日の曜日を呼び出し、分岐で毎週土曜日だけ選択出来るようにする。

トップページから移動出来るページは、食品の登録・近況・体重登録・登録内容の修正がある。また、ログアウトはトップページでのみ実行出来る。

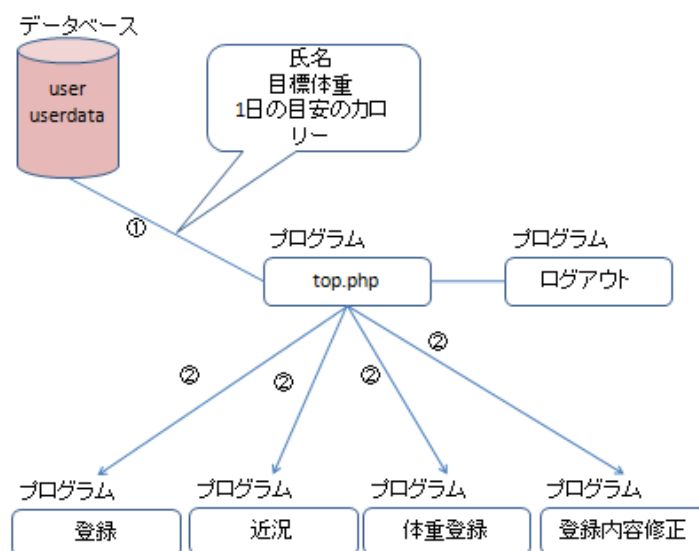


図 4-3-3-1: トップ画面のブロック図

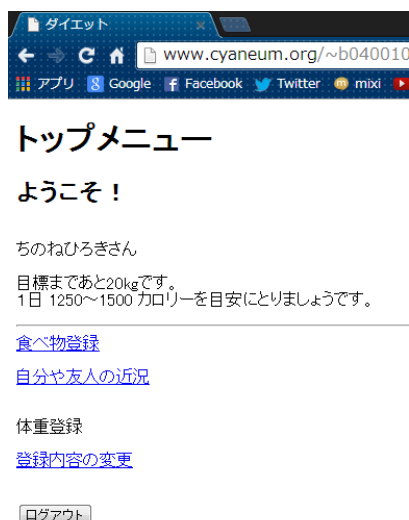


図 4-3-3-2: トップ画面

表 4-3-3-3: Top. php

```

<?php
session_start();
// ログイン状態のチェック
if (!isset($_SESSION["uname"])) {
    header("Location: login.php");
    exit;
}

define("PASSWORD", "pass");
$message = "";
if(isset($_COOKIE["TEST_COOKIE"]) && $_COOKIE["TEST_COOKIE"] != ""){
    $_SESSION["TEST"] = $_COOKIE["TEST_COOKIE"];
}

if(isset($_SESSION["TEST"]) && $_SESSION["TEST"] != null && md5(PASSWORD) ===
$_SESSION["TEST"]){
    $message = "Login success";
}else{
    session_destroy();//セッション破棄
    header("Location:login.php");
}
?>

<!DOCTYPE HTML>
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8"

```



```

        name="viewport" content="width=device-width,
        initial-scale=1, maximum-scale=2" />
</head>
<body>
<h1>トップメニュー</h1>

        <?php
mysql_connect('localhost', 'b040010', '*****');
mysql_select_db('b040010');
$uid = $_SESSION["uname"];

$sql = "select
taiju.tday, userdata.uname, userdata.username, taiju.taiju, userdata.mokutai, use
rdata.mokukc, userdata.mokukec2 from taiju inner join userdata on taiju.uname =
userdata.uname and taiju.uname = '$uid' order by tday desc limit 1;";
$result = mysql_query($sql);
$row = mysql_num_rows($result);

        if($row == 0) {
            echo "<p>該当データがありません。</p>";
        } else {
            mysql_query(sql2);
            while($row = mysql_fetch_array($result)) {
                echo "<p>";
                echo "<h2>ようこそ! </h2><br />";
                echo $row["username"];
                echo "さん<br /><br />";
                echo "目標まであと";
                echo $row["taiju"] - $row["mokutai"];
            }
        }
    }
}

```

```

        echo "kg です。<br />";
        echo " 1 日 ";
        echo $row["mokukc"];
        echo "～";
        echo $row["mokukc2"];
        echo " カロリーを目安にとりましょうです。";
        echo "</p><hr />";
    }}
?>

```

```

<a href="diet.html">食べ物登録</a><br /><br />
<a href="showdiet.php">自分や友人の近況</a><br /><br />
    <?php
        $youbi= date("w");
        if($youbi == 6){
            echo "<p><a href=taiju.html>体重登録</a></p>";
        }else { echo "<p>体重登録</p>";}?>
<a href="syusei.php">登録内容の変更</a><br /><br />
<form action="logout.php" method="post">
<p><button type="submit" name="logout">ログアウト</button></p>
</form>
</body>
</html>

```

#### 4-3-5. 食べた食品の登録

入力の手間を省くために、食べた日付・時間を now 関数で取得する。撮影した写真は、ユーザ ID と写真名をつなげて、ddb のデータベースに保存し、画像をフォルダに

保存する。画像は、move\_uploaded\_file(\$im, \$pla)でフォルダに移動する。Ddb のデータベースに保存した画像の名前と、フォルダに保存した画像の名前が一致するようにする。

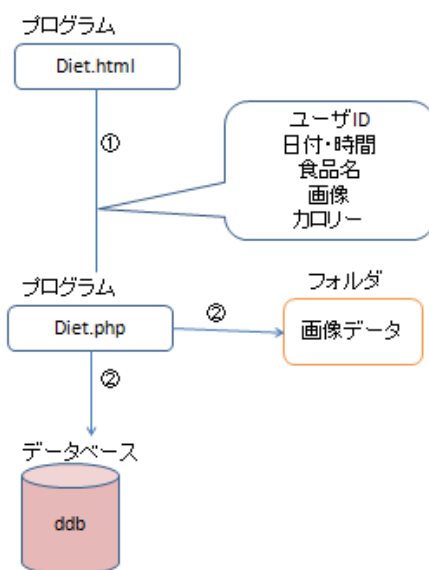


図 4-3-5-1: 食べた食品登録のブロック図

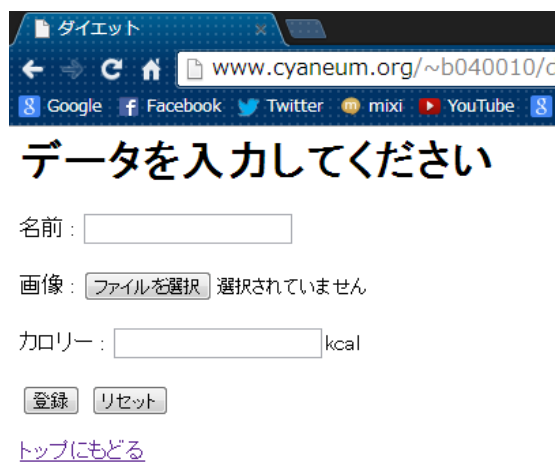


図 4-3-5-2: 食品の登録

表 4-3-5-3: Diet.html

```
<?php
session_start();
// ログイン状態のチェック
if (!isset($_SESSION["uname"])) {
```

```

header("Location: login.php");
exit;
}

<!DOCTYPE HTML PUBLIC>
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8">
</head>
<body>
<h1>データを入力してください</h1>
<form action="diet.php" method="post" enctype="multipart/form-data">
<p>食べ物名 : <input type="text" name="nam" size="20"></p>
<p>画像 : <input type="file" name="upfile" size="20"></p>
<p>カロリー : <input type="text" name="cal" size="20">kcal</p>
<p><input type="submit" value="登録">
<input type="reset" value="リセット"></p>
<a href="top.php">トップにもどる</a>
</form>
</body>
</html>

```

表 4-3-5-4:Diet.php

```

<?php
session_start();
if (!isset($_SESSION["uname"])) {
    header("Location: login.php");
}

```

```

}
?>
<!DOCTYPE HTML PUBLIC>
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8">
</head>
<body>

<?php
extract($_POST);
mysql_connect('localhost','b040010','*****');
mysql_select_db('b040010');

$uid = $_SESSION["uname"];
$dy = date("y-m-d");
$tm = date("H:i:s");
$im = $_FILES["upfile"]["tmp_name"];
$imn = $_FILES["upfile"]["name"];
$pname = $_SESSION["uname"].$nam.$FILES["upfile"]["name"];
$pla = "pic/" . $pname;

if ($_FILES["upfile"]["error"] == UPLOAD_ERR_OK && is_uploaded_file($im)) {
    move_uploaded_file($im, $pla);
    chmod($pla, 0644);
    echo "正常にアップロードされました。";
} else {

```

```

        echo "アップロード失敗しました。";
    }

    $sql = "insert into ddb values('$uid', '$dy', '$tm', '$nam', '$pname',
    '$cal',0,0)";
    $result = mysql_query($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 "<a href = diet.html>戻る</a>";
        echo "</p>";
    } else {
        echo "データベースに書き込み完了";
        echo "<a href=top.php>トップにもどる</a>";
    }
?>

</body>
</html>

```

#### 4-3-6. 登録された食品を見る

4-3-5 項で登録された内容を確認するページを作成する。User と ddb のデータベースから名前・日付・時間・食品名・画像・カロリーを参照し、表示させる。Ddb のデータベースに保存してある画像の名前をフォルダの中から一致する画像を表示させる。

inner join で userdata と ddb のデータベースをユーザ ID を鍵として繋げる sql 文を使用したため、userdata から氏名を参照してくる事が出来た。

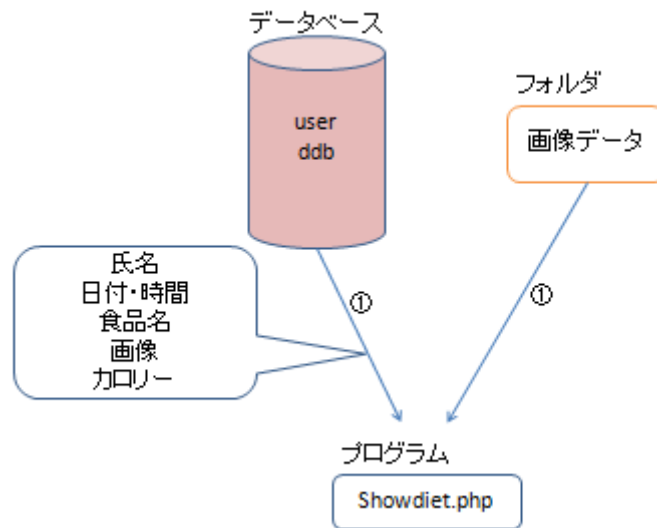


図 4-3-6-1:登録された食品を見るブロック図



図 4-3-6-2:登録された食品を見る

表 4-3-6-3:Showdiet.php

```

<?php
session_start();
// ログイン状態のチェック
if (!isset($_SESSION["uname"])) {
    header("Location: login.php");
  
```

```

    exit;
}
?>
<!DOCTYPE HTML PUBLIC>
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8">
</head>
    <body>
<a href="top.php">トップにもどる</a>
<?php
mysql_connect('localhost','b040010','*****');
mysql_select_db('b040010');
extract($_POST);

$sql= "select
userdata. uname, userdata. username, ddb. day, ddb. tim, ddb. name, ddb. img, ddb. kcal, d
db. jtai from ddb inner join userdata on ddb. uname = userdata. uname";

$sql2 = "update ddb set jtai=jtai+1 where uname = '$uid' and name = $na";
$prs = $_POST["prs"];
$uid = $_POST["una"];
$na = $_POST["na"];
$prs = $_POST["prs"];

$sql= "select
userdata. uname, userdata. username, ddb. day, ddb. tim, ddb. name, ddb. img, ddb. kcal, d
db. jtai from ddb inner join userdata on ddb. uname = userdata. uname";

```



```

$sql2 = "update ddb set jtai=jtai+1 where uname = '$uid' and name = $na";
$result = mysql_query($sql);
$rows = mysql_num_rows($result);

if($prs == 1){mysql_query($sql2);}

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["username"];
        echo " <br /> ";
        echo $row["day"];
        echo " ";
        echo $row["tim"];
        echo " ";
        echo "<br />";
        echo $row["name"];
        echo"<br />";
        echo $row["img"];
        echo "<br />";
        echo "<img width=240 src=¥"pic/" . $row["img"]. "¥">";
        echo "<br />";
        echo $row["kcal"];
    }
}

```

```

        echo "kcal";
        echo "<br />";
        echo "食べ過ぎ報告";
        echo $row["jtai"];
        echo "件          ";
        echo "<form action=¥"showdiet.php¥" method=¥"post¥">";
        echo "<input type=¥"hidden¥" name=¥"prs¥" value=¥"1¥">";
        echo "<input type=¥"hidden¥" name=¥"na¥" value=¥""";
        echo $row["name"];
        echo "¥">";
        echo "<input type=¥"hidden¥" name=¥"una¥" value=¥""";
        echo $row["uname"];
        echo "¥">";
        echo "<input type=¥"submit¥" value=¥"食べすぎ¥">";
        echo "</form>";
        echo "</p><hr />";

    }

}

?>

</body>
</html>

```

#### 4-3-7. 体重を登録

Top.php で分岐を作り、曜日を取得し、もし土曜日なら taiju.html を選択出来るようにする。トップ画面で体重登録が選択出来る場合は体重を登録する事が出来る。

体重の項目にデータを入力し、登録ボタンを押すと taiju のデータベースに登録が完了する。

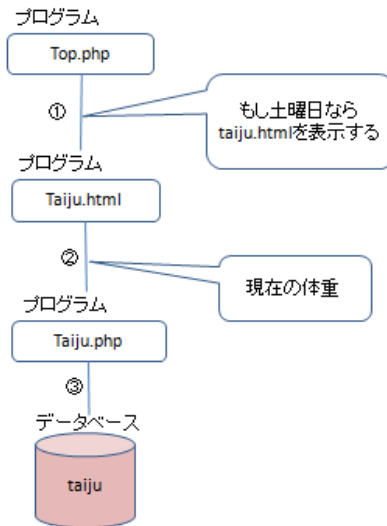


図 4-3-7-1: 体重登録のブロック図



図 4-3-7-2: 体重登録

表 4-3-7-3: Taiju.html

```
<?php
session_start();
if (!isset($_SESSION["uname"])) {
    header("Location: login.php");
```

```

}
?>
<!DOCTYPE HTML>
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8">
</head>
<body>

<h1>体重登録</h1>
<form action="taiju.php" method="post">
<p>体重 : <input type="text" name="tai" size="10">Kg</p>
<p><input type="submit" value="登録"></p>
<a href="top.php">トップにもどる</a>
</form>

</body>
</html>

```

表 4-3-7-4:Taiju.php

```

<?php
session_start();
if (!isset($_SESSION["uname"])) {
    header("Location: login.php");
}
?>
<!DOCTYPE HTML>

```

```
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8">
</head>

<body>

<?php
extract($_POST);
mysql_connect('localhost','b040010','*****');
mysql_select_db('b040010');

$uid = $_SESSION["uname"];
$dy = date("y-m-d");
$sql = "insert into taiju values('$dy','$uid','$tai')";
$result = mysql_query($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 "<a href = diet.html>戻る</a>";
    echo "</p>";
} else {
    echo "書き込み完了";
    echo "<a href = top.php>トップにもどる</a>";
}
}
```

```
?>
```

```
</body>
```

```
</html>
```

#### 4-3-7. アカウントの内容を変更する

User と userdata のデータベースから内容を読み込み、テキストに表示させる。修正した項目の内容を各データベースへ上書き保存する。inner join で user と userdata を結合し、現在の自分のデータを読み出す。各項目変更が完了し、送信ボタンを押すとパスワードは user へ、氏名と目標体重は userdata へ上書き保存をする。

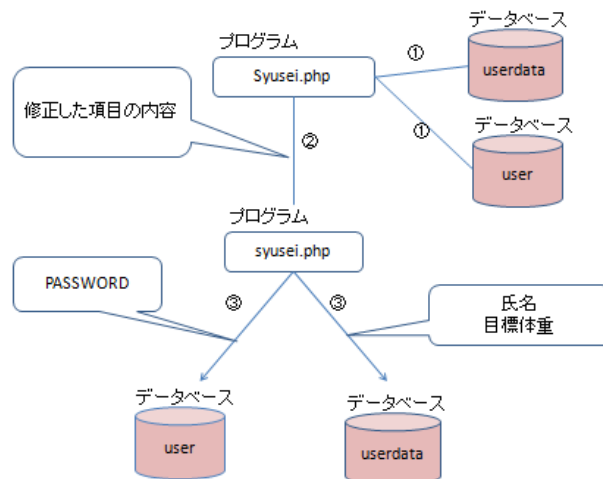


図 4-3-7-1: 登録内容変更のブロック図



図 4-3-7-2: 登録内容の変更

表 4-3-7-3:Syusei.php

```

<?php
session_start();
// ログイン状態のチェック
if (!isset($_SESSION["uname"])) {
    header("Location: login.php");
    exit;
}
?>
<!DOCTYPE HTML PUBLIC>
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8">
</head>

<body>

<?php
mysql_connect('localhost', 'b040010', '*****');
mysql_select_db('b040010');
extract($_POST);
extract($_GET);
$ume = $_SESSION["uname"];

if($una<>"") {
$sql = "update userdata set uname = '$una', username = '$unm', mokutai =

```

```

'$mok', mokukc = '$mok * 25', mokukc2 = '$mok * 30';

$result = mysql_query($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 "<a href = diet.html>戻る</a>";
    echo "</p>";
} else {
    echo "データベースに書き込み完了";
    echo "<a href = top.php>トップにもどる</a>";
}
exit;
}

$sql = "select * from userdata where uname = '$ume'";
$result = mysql_query($sql);
$rows = mysql_num_rows($result);
if($rows == 0){
    echo "<p></p>";
}
else{
    while($row = mysql_fetch_array($result)){
        echo "<h1>登録内容の変更</h1>";
        echo "<form action = ¥"syusei.php¥" method = ¥"post¥">";

        echo "<p>ユーザ ID: ";

```



```

        echo $row["uname"];
        echo "</p>";

        echo "<p>氏名: <input type='text' size='20' name='unm'
value='";
        echo $row["username"];
        echo "' size = '1'"></p>";

        echo "<p>目標: <input type='text' name='mok' value='";
        echo $row["mokutai"];
        echo "' size = '1'"></p>";

        echo "<input type='hidden' name='una' value='";
        echo $row["uname"];
        echo "'>";

        echo "<p><input type='submit' value='送信'></a>";
        echo "<p><a href = top.php>戻る</a>";
        echo "</form>";

    }
}
?>

</body>
</html>

```

#### 4-7-8. ログアウトする

ログアウトボタンが押されたら\$\_sessionの中を空にし、Cookiesを削除する。

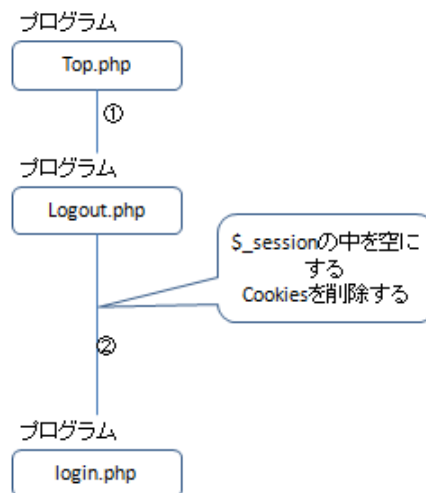


図 4-7-8-1: ログアウトのブロック図

表 4-7-8-2: logout.php

```
<?php
session_start();
if(isset($_POST['logout'])) {

// セッション変数を全て解除する
$_SESSION = array();

// セッションを切断するにはセッションクッキーも削除する。
// Note: セッション情報だけでなくセッションを破壊する。
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
};
```

```
}

//セッションを破壊してリダイレクト
session_destroy();

}

?>
<!DOCTYPE HTML PUBLIC>
<html lang="ja">
<head>
    <title>ダイエット</title>
    <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8">
</head>
<body>
    <p>ログアウト完了</p>
    <a href = login.php>ログイン画面に戻る</a>
</body>
</html>
```

## 5章 評価と考察

### 5-1. 動作検証

どのページでも、ログイン状態を確認しているためログインを行っていない場合は必ず login.php が表示される。现阶段では、ID を持っていないため新規登録を選択してユーザ ID とパスワードの設定を行う。



図 5-1-1: login.php

新規登録を選択すると、新規登録画面に移動する事が出来る。新規登録では ID・PASSWORD・名前・目標体重を入力する。入力が完了したら登録ボタンを選択する。



図 5-1-2: shintouroku.html

図 5-1-2 で登録した ID と PASSWORD を入力しログインする。PASSWORD は見えないようになっているので、誤入力に気をつける。チェックボックスにチェックを入れログ

インすると、次回から自動的にログイン出来る。



図 5-1-3: login. php

ログインが完了するとトップ画面を表示される。トップメニューには、新規登録で入力された氏名・現在の体重から目標体重を引いた数・目標体重から算出した 1 日の目安摂取カロリーのデータが参照されている。

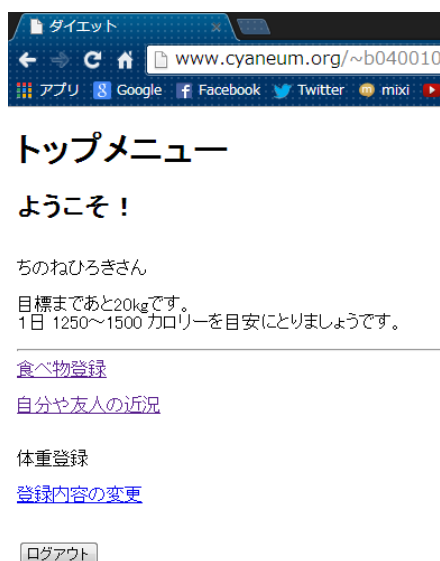


図 5-1-4: top. php

トップ画面から食べ物登録を選ぶと図 5-1-5 が表示される。入力する項目は、食べ物の画像を用意出来る場合は画像の選択だけだ。画像が用意できない場合は、名前とカロリーを入力する。ファイルを選択のボタンを押すと、パソコンならば、画像フォルダが開かれる。スマートフォンであった場合は写真の選択とカメラを起動が選択出来

る。

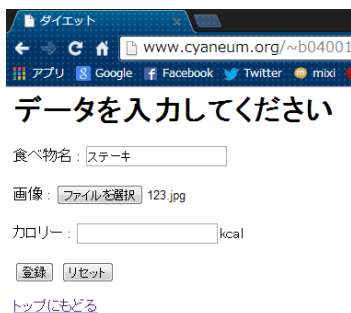


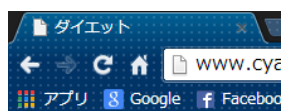
図 5-1-5:diet.html

入力が終わり登録ボタンを押すと登録が完了し、トップ画面に戻る。次に登録した内容を見るために、自分や友人の近況を選択する。すると図 5-1-6 が表示された。日付や時間は入力していないが表示されている。



図 5-1-6:showdiet.php

実装で、体重の登録は毎週土曜日に表示するように設定した。体重の記録は土曜日まで待ち行う。体重の登録が完了するとトップ画面の目標までの kg 数が増えた。



## 体重登録

体重:  Kg

[トップにもどる](#)

図 5-1-7:taiju.html

トップ画面から登録内容の変更を選択すると、ユーザの情報が参照されて表示される。指名を「ちのねひろき」から「ちのね」に変更する。また、目標体重を「50」から「60」に変更し登録する。



## 登録内容の変更

ユーザID: chinone

PASSWORD:

氏名:

目標:

[戻る](#)

図 5-1-8:syusei.html

登録内容の修正が完了し、トップ画面に映ると名前が変わり、現在の体重と目標体重の差も修正されている。



## トップメニュー

ようこそ!

ちのねさん

目標まであと30kgです。  
1日 1000~1200 カロリーを目安にとりましょうです。

[食べ物登録](#)

[自分や友人の近況](#)

[体重登録](#)

[登録内容の変更](#)

図 5-1-9:top.php

トップ画面でログアウトを実行する。すると、図 5-1-10 が表示された。ログアウトが完了した状態で top.php にアクセスをすると、ログインされていないため login.php が表示された。



図 5-1-10:logout.php

## 5-2. 評価

4 章で行った実装の評価を行う。また実際にアプリケーションを使用してもらい評価をしてもらう。

### 5-2-1. 実装の評価

3 章で提案した解決法に基づいた機能の評価をする。

表 5-2-1-1:評価

| 番号  | 解決法                 | 実装 |
|-----|---------------------|----|
| (1) | 写真で記録を簡単にする         | ○  |
| (2) | 体重の登録を 1 週間に 1 回にする | ○  |
| (3) | グループ機能を取り付ける        | ×  |

#### (1) 写真で記録を簡単にする

設計目標は写真をアップロードし表示させる事である。写真が表示される事によって、他の項目を入力しなくても分かるようになった。同じ名前の画像がアップロードされると区別が出来ず最新の画像が表示されるようになってしまった。しかし、画像の名前にユーザ ID を入れる事によって改善した。



## (2) 体重の登録を1週間に1回にする

設計目標は、1週間1回だけ体重登録が選択出来るようにする事である。トップ画面でIf文により分岐を作り、ページの表示を操作する事が出来た。実装で、記録日を土曜日としていたが今後はユーザによって設定出来るようにする事が必要だ。

## (3) グループ機能を取り付ける

登録されたユーザの名前やユーザIDで検索をし、友人を探し自分の好きな人とグループを作れる機能を考えていた。他のユーザが登録した内容を表示する事が出来たが、内容を表示する制限をする事が出来なかった。この機能が実現する事によってグループを組む事や、一人での使用も可能になる。また、SNSのように友人とのコミュニケーションをとる手段にする事が出来る。

### 5-2-2. モニターの評価

モニター調査として6人に実際に本研究で作成したアプリケーションの使用と評価をしてもらった。アプリケーションの使用期間は2週間で、この6人は全員お互いの体型を知っている友人同士である。なお、使用期間が短かったため、体重が大きく変化したユーザはいなかった。

表 5-2-1-1：モニターによる評価

|      | 良い点  | 悪い点                        |
|------|--|----------------------------|
| ユーザ1 | 1日に摂取しても良いカロリーの目安が表示されていて、便利だった。                 | 食事の度に写真を撮るのが面倒だった。         |
| ユーザ2 | 自分が食べた物を登録する事で友人に見られているので、カロリーの摂取しすぎに気をつける事ができた。 | 間違えて食べ物登録した時のために修正機能がほしい。  |
| ユーザ3 | 自分や友人が食べた物がわかり、SNSのようで楽しかった。                     | 写真を投稿するのが楽しくなり、つい食べ過ぎてしまう。 |
| ユーザ4 | 自分の登録した写真を見て、食                                   | 体重を入力出来る曜日が決ま              |

|       |   |   |
|-------|---|---|
|       | 生活の乱れがわかって食生活を改めようと思った。                                   | っているので、その曜日に体重の入力ができなかったり入力し忘れたりすると時に不便だった。         |
| ユーザ 5 | ユーザが自分の体型を知っている友人だったため、知らない人に食事を見られるのよりカロリー摂取に気をつけようと思った。 | せっかく食べ物登録の時にカロリーを入力しているのに、今日あとどれくらい摂取していいのが表示してほしい。 |
| ユーザ 6 | 今までカロリーを気にしないで食事をしてきたが、このアプリケーションを使って初めてカロリーを気にするようになった。  | 登録した食べ物を見るときに、ユーザごとに表示出来るようにしてほしい。                  |

### 5-3. 考察

本研究で実装したアプリケーションを実際に使用してもらい、良い点・悪い点の評価をしてもらった。良い点から、写真をアップロードする事によって自分の身近な人に食生活を見られるので規則正しく、また食べすぎずに食事をするように意識させる事が出来た。しかし、悪い点からは食べ物の登録内容がユーザごとに表示させてほしい、食べ物名・カロリーを入力するのが面倒であったと評価された。

友人に自分の登録内容を表示させる事だけを重視していたために、ユーザごとの表示に気がつく事が出来なかった。人に見られる事と同時に、自分の食べた物だけでも表示される事によって、自分がどのような食生活をしているかに気付き、食生活の改善に繋がるため今後の検討が必要である。食べ物登録では、入力する項目を減らすと述べたが、画像が用意出来ない場合に備え、あえて食べ物名とカロリーの入力項目を設置した。しかし、設置した事でユーザが項目を埋めなければいけないと思わせてしまった。画像が準備されていない時に、項目の入力が可能な状態にする事で登録がスムーズに行えただろう。

## 6章 まとめと今後の課題

### 6-1. まとめ

本研究の目的は、今の体の状況を把握するために食べ物や体重の記録・管理、家族や友人などの身の回りの人と記録を共有する事によって体重維持、減量を長く続ける事の手助けにする事を目指した。4つの既存のWebサービスを調べ、機能を分析し、実際に使用して既存のWebサービスの問題点を見つけ、解決するために必要な機能を検討し、設計から実装を行い、実装の評価と実装したアプリケーションを実際に使用してもらった評価をした。

問題点として、食べた食品の名前・カロリー・量を自分で調べ打ち込む事、また毎日の体重の登録は繁雑である事と、コミュニティの利用によるダイエットの継続は難しい所があげられる。この問題を解決するために、食べた食品の写真を撮り記録し、体重の登録は1週間に1回登録出来る仕組みを構築した。この仕組みにより、写真から食品名と量が見て分かるようになった。体重の記録では、毎日同じ体重を打ち続けるよりも高いモチベーションを維持出来るようになった。また自分の身の回りの人とグループを作る事によって、食生活を見られるので規則正しく、また食べ過ぎに注意するように食事をとる意識をさせる事が出来た。

### 6-2. 今後の課題

今回実装する事が出来なかった機能の完成を目指す。また、モニター調査で明らかになった所の改善をする。以下の事を今後の課題とし、理想のアプリケーションを実現するために、改善に向け検討する。

#### 6-2-1. グループ機能

グループ機能の完成を目指す。グループ機能を使用する事で、ユーザが増えた場合にさまざまなグループを作る事が出来るようにする。また、登録した内容は同じグループの人だけが見られるようにする必要がある。

#### 6-2-2. 近況の表示方法

モニター調査で明らかになった、登録された食品をユーザごとに表示される機能の

完成を目指す。この機能が実現する事によって、自分が登録した内容だけを見る事が出来る。自分がどのような食生活をしているかに気付き、食生活の改善を図るが出来る。

### 6-2-3. データをグラフによる視覚化

食べた食品の登録時間から食事を取る時間のグラフや週ごとの体重のグラフを表示させ、分析するための機能の完成を目指す。この機能が完成する事によって、自分が良く食事をする時間帯を知る事が出来る。また、今まで登録された体重をグラフ化し、比較する。先週よりも体重が重くなっている場合は、先週食べた物や食事の回数を表示する事で分析する事が出来る。

## 参考文献

- (1) 国立循環器病研究センターの循環器病情報サービス

<http://www.ncvc.go.jp/cvdinfo/pamphlet/obesity/pamph03.html#anchor-3>

2012年12月12日

- (2) 厚生労働省

<http://www.mhlw.go.jp/topics/bukyoku/kenkou/seikatu/himan/about.html>

2012年7月21日

- (3) MACROMILL×InMobi の共同調査

[http://www.macromill.com/r\\_data/20120726diet\\_smartphone/](http://www.macromill.com/r_data/20120726diet_smartphone/)

2012年7月20日

- (4) 東京都病院経営部

<http://www.byouin.metro.tokyo.jp/eiyou/himan.html>

2012年12月1日

- (5) Girl's Diet

<http://girls-diet.jp/>

2012年7月4日

- (6) あすけん

<http://www.asken.jp/>

2012年7月24日

- (7) 鈴木 憲治・山田 直明・山本 義之・浅野 仁・櫻井 雄大・安藤建一

PHP 逆引きレシピ 第2版 (PROGRAMMER'S RECIPE)

翔泳社 2013年

- (8) 森 卓郎 運動指導者が断言! ダイエットは運動1割・食事9割

インプレスコミュニケーションズ 2013年

## 謝辞

本研究を行うにあたり、ご指導いただいた渡辺恭人准教授に大変感謝しております。論文の構成や執筆にいたるまでたくさんのご指導をしていただいたおかげで、本研究を進める事が出来ました。

渡辺恭人准教授には、入学してから4年間お世話になりました。1年生の頃は、研究基礎や情報基礎で発表の技法やITの基礎をご教示いただきました。2年生からは、ゼミナールに参加させていただくようになり、プログラミングやデータベースを学んだ事によって、本研究の実装を実現する事が出来ました。心より感謝いたします。

最後に、私の卒業論文に関わって下さった方全員に感謝を述べさせていただき謝辞とさせていただきます。ありがとうございました。