

# 2 0 0 8 年度卒業研究

「携帯電話による出席確認および、  
位置情報取得サービスの検討」

担当教員： 渡辺 恭人

0540202

牛濱 優

## 目次

### 概要

1. 背景・目的	4
1-1 背景	4
1-2 目的	4
2. 大学の現状	5
2-1 千葉商科大学の現状	5
2-2 札幌大学における出席確認	5
2-3 フリーソフトでの出退勤ソフト	6
2-4 現在のシステムでの問題点	11
2-5 問題点をふまえて	12
3. 解決策の検討	13
3-1 解決へのアプローチ	13
3-2 解決法の提案	15
4. 設計	16
4-1 目標	16
4-2 今回のプログラムにおいて必要な要素	16
4-3 機能要素	19
4-4 テーブルの設計	23
5. 実装	24
5-1 実装環境	24
5-2 MYSQL へのテーブル設定	24
5-3 個人の情報の登録	26
5-4 実行結果	35
5-5 動作	39
5-6 考察	39
6. まとめと今後の課題	41
6-1 まとめ	41
6-2 今後の課題	41

参考文献.....	42
謝辞.....	43

## 概要

日常生活において、大学の授業では出欠管理が重要な業務あることが多い。現在は、出席カード等の紙媒体を使用しており、配布、記入、カウントするなどに時間を要し、授業時間や業務時間を無駄に使っている。また、学内での先生の居場所がわからず、なかなか面会できないことがあります。

そのような場面で先生を探す時間や授業中に出席を取る際にわざわざ出席カードを配る時間を減らしたい。携帯電話という汎用的な機器の機能を利用して先生、学生を識別、認証することによって、出欠管理とそれを応用した位置管理機能とサービスを検討し、構築、提供を目指します。

### 1. 背景・目的

#### 1-1 背景

大学の授業では出席状況の確認などで日常的に行われているが授業によっては出席が大きく成績に関係する場合があります、出席確認が重要となります。

現在の出席確認方法として出席カードや点呼による出席確認を行っている授業などもあります。

出席カードの場合、毎回カードを人数分配布するか、または出席カードを置いておいて自分たちに取りに来させ後で提出してもらい回収する。

毎回配布と回収で5分から10分くらいの時間が出席確認に費やされているまた点呼の場合でも同様である程度同じ時間が費やされているがしかし、大人数の場合では出席を集計するだけでも膨大な時間が費やされてしまいます。

また、点呼の場合ではその生徒の顔を確認などができますが出席カードを使って出席確認をする場合他の人を使って不正を行うことができてしまいます。

#### 1-2 目的

出席確認や回収、集計の手間や時間を縮小し、不正も起こりにくくする出席確認方法について検討する。

汎用的な機器及びソフトウェアを利用してできるだけ低コストシステムを実現させていくことを目的とします。

## 2. 大学内の現状と問題点

### 2-1 千葉商科大学の現状

現在の大学では、出席カード（図 2-1）を使って出席確認が多い。

千葉商科大学での出席

- ・履修登録時のデータをもとに生徒の一覧を作成
- ・出席カードや点呼を使って出席確認
- ・出席状況は出席カードに印や色を変えるなどして判断
- ・出席カードを使っての不正が可能
- ・先生が出席カードを無くしてしまう場合がある

さらに、学生の視点としてこの広い学内で友人などに会う時に8つの会館やたくさんの教室、また休憩所など多くの部屋などがあります。そういった中で出会う場合場所を特定することはとても困難である。

氏 名					
学 科		学 年	組	番	
年	月	日	曜	時 限	教 室
科 目				担 当 者	先 生

千 葉 商 科 大 学

図 2-1 出席カード

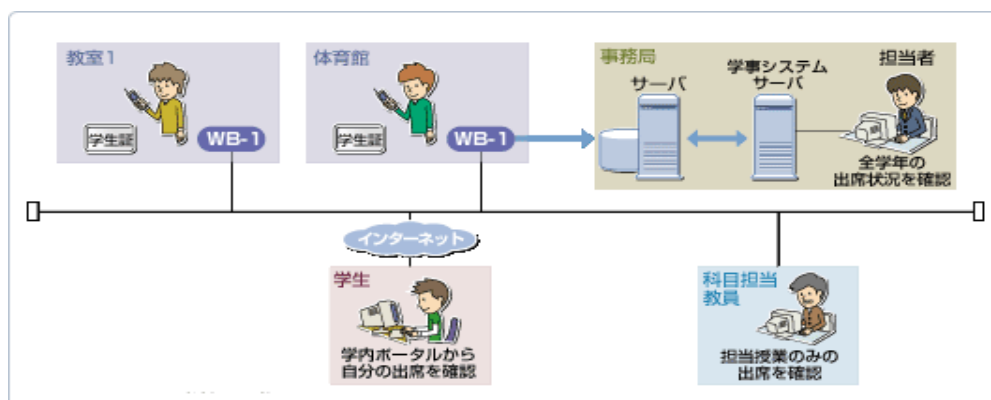
### 2-2 札幌大学における出席確認

まず、全国の大学での出席確認方法について調査した。

調査してみた結果、札幌大学などで携帯電話の FeilCa 機能を使っての出席確認を導入している大学があった。出席確認の方法として（図 2-2）に示す。

札幌大学では、出席確認において Felica チップを内蔵したものと IC チップを内蔵したカードの両方を使うことができる。また使っているリーダーは取り外しが可能で導入されていない部屋などに移動した際や、野外などでも使うことができるものでした。

図 2-2 札幌大学での利用事例 (<http://www.docomo-sys.co.jp/cstudy/cstudy10.html>)



他大学での実用例などを調べてみた結果多くの大学で同じような携帯電話を使っての出欠確認を行っています。

例などは出てはいませんが、どれも同じように教室に Felica ポートを設置してそこに携帯電話をかざすことによって出欠確認をしている。

## 2-3 フリーソフトでの出退勤ソフト

### JSTTimeRecorder

出席ではありませんが会社などでの出勤や退社などを Felica 機能を使って出勤や退社の時間などを管理するものもありました。(図 2-3-1) このソフトウェアは、最初に Felica の id を登録してその後、登録画面にて名前や部署などを入力(図 2-3-3)したあと会社などで出勤時などにソフトウェアを起動させておいたパソコンを置いておきそこに Felica をかざすことによってその人の出勤時間や退社した時間などを記録することのできるソフトウェアである。またこのソフトウェアではデータを自動的にエクセルとして出力することもできるといった利点があります。

### 2-3-1 JStimerecorder の起動画面

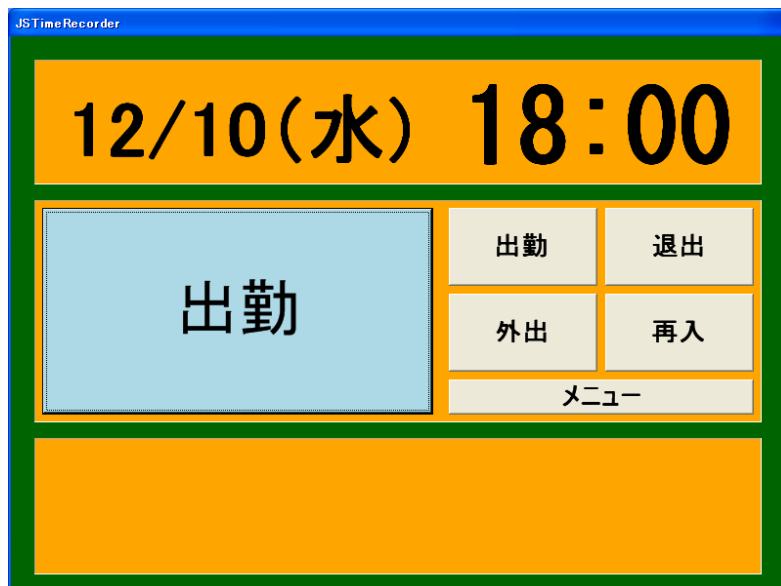
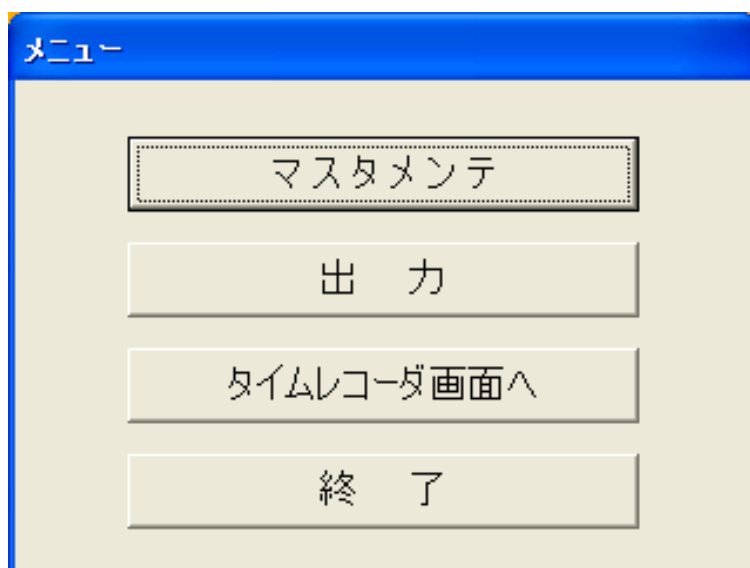


図 2-3-2 JStimerecorder のメニューボタン内のマスタメンテ内で情報の登録をすることができる



### 2-3-3 マスタ内での登録

図 2-3-3 マスタ内での個人情報登録

この中にある新規登録で登録することによって登録することができるものである。

2-3-4 登録した状態でのタイムレコーダー画面は

図 2-3-4 登録された画面



登録されていない feilca カードだった場合の画面

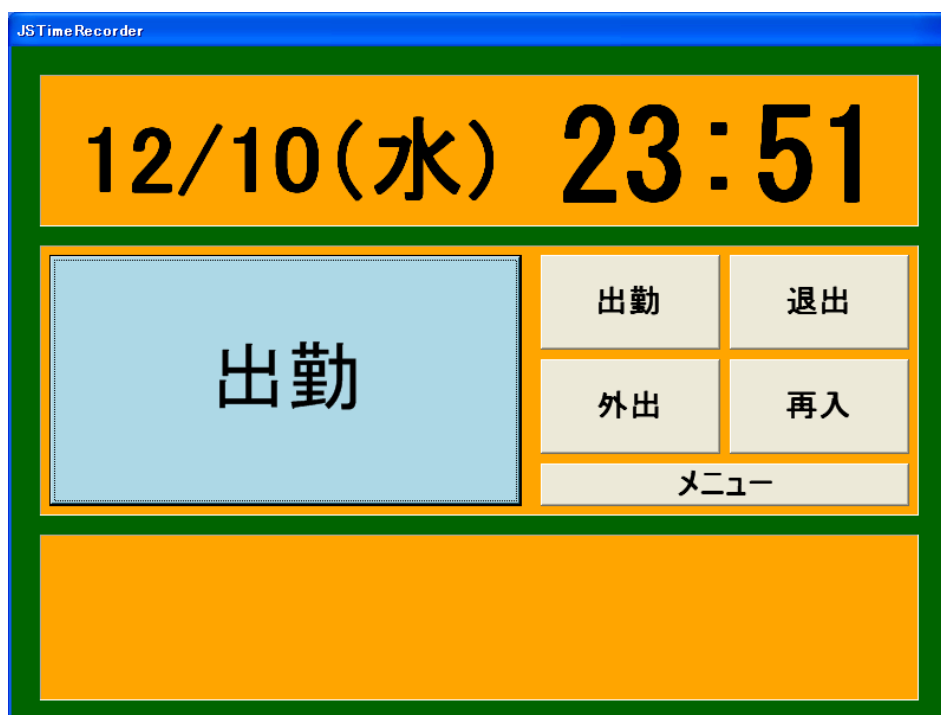


図 2-3-5 登録されていない画面

こうした情報を表示することにより誰が来て何時までいたのかなどを表示してまたメニューでの登録する場合には下の図のようなパスワードの入力画面になり管理者以外の人には操作できません。

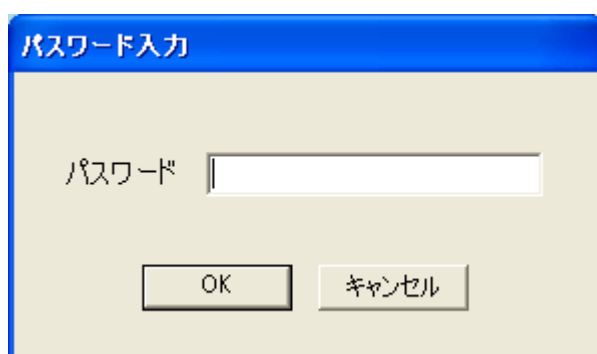


図 2-3-6 パスワード入力画面

またシフトに対応して、シフトを新しく設定することも可能です。

シフト変更の場合

図 2-3-7 シフト変更画面

また下の図 2-3-8 のようにエクセルのファイルとして出退勤状況を出力することができ、紙の資料として管理も可能です。その結果を図 2-3-8 に示す。

図 2-3-8 勤怠明細票

### 勤 怠 明 細 表

(2008 年 12 月 1 日～2008 年 12 月 31 日)

グループ	01	営業
氏 名	01	牛浜 優

日付	シフト	出勤	外出	再入	退出	備考
12/01(月)						
12/02(火)						

12/03(水)						
12/04(木)						
12/05(金)						
12/06(土)						
12/07(日)						
12/08(月)						
12/09(火)						
12/10(水)	日勤	21:35				[出]12/10 21:35
12/11(木)						
12/12(金)						
12/13(土)						
12/14(日)						
12/15(月)						
12/16(火)						
12/17(水)	日勤	21:33				

## 2-4 現在のシステムでの問題点

出席票を使う場合の問題点

- ・紛失してしまう場合がある。
- ・不正が可能
- ・回収や集計に時間がかかる。

札幌大学の問題点

- ・出席確認の Felica ポートが移動可能なためポートの管理が必要
- ・Felica ポートを使つての位置情報が取得することはできない

JSttimerecorderno の問題点

- ・登録のときに Felica を別のものを使って登録した場合にも同じように使えてしまう
- ・パスワードの設定が簡単に変えることができってしまうためセキュリティに問題

## 2-5 問題点をふまえて

出席カードの問題点は不正や紛失、回収といった部分は札幌大学の出欠登録でそういった部分はできるし、集計に関してもデータをサーバーに送っているため集計も一気にすることができ時間も出席カードに比べたら時間もかからない。

JSttimerecorder では登録やエクセルとして出力できるといった利点もあるが管理といった部分に問題があります。

出席確認として JSttimerecorder の名前登録と札幌大学の出欠登録を参考にし、これからのシステムを制作していく。

### 3. 解決法の検討

#### 3-1 解決へのアプローチ

紙や点呼による確認や記録は時間がかかるだけでなくその後の管理や集計の手順が複雑で不正も起きてしまいます。また

出席において出席カードといったものを使い出席を確認する場合にはその紙を使つての確認でしかありませんが、その紙を使わずに、データとして確認が取れるようになれば最後の集計においても簡単にデータを反映させてしまえばいいものなのですぐにデータを出すこともできるのである。

また、出席カードにおける不正をさせないために普段学生が持ち歩き、個人の識別ができるデバイスの利用として考えたのが携帯電話であり、そのデバイスの情報をデータとして WEB に送ることによって出欠の管理が簡単に利用できるといった部分もある。

そして、個人の携帯電話ならば不正も困難でその人自身が来ないと授業の出欠情報を登録することができない。また、遅刻などについても授業の決まった時間までに携帯電話をかざさなくてはいけないので授業への遅刻を少なくするといった意味でも効果がある。

なぜ携帯電話なのかというと今の携帯電話には Felica 機能がほとんどの携帯電話に普及しているからです。

Felica とは、ソニーが開発した非接触 IC カード技術です。

FeliCa では一枚のカードで複数種類のデータを管理することができるため、電子マネーや公共交通機関のプリペイドカード、社員証、学生証、入退室管理の ID カードなどの機能を併せ持ったカードを作ることができます。これらのデータは FeliCa チップの中で独立して管理されており、個別のアクセス権を設定することが可能で、複数のサービス間で安全な相互運用が可能となっています。

Felica 機能は今ある Docomo や au、SoftBank といった携帯ほとんどに普及していてどの機種でも同じように使えるからです。

Felica は USB でパソコンに接続することにより Felica のデータを取得することができ、自宅などいろいろなところで扱うことができるといった点で Felica を選

択しました。

## 3-2 解決法の提案

### 3-2-1 アナログな方法からデジタルな方法への移行

現状ではアナログな部分にデジタル部分を加えることによって改良を行っている先生も存在する。

#### ① 点呼+Excel

ある先生は履修名簿をもとに点呼をとり、出欠を自分の PC の表計算ソフト Excel に記録をしている。欠席 0 出席 1 遅刻 0.5 といった数値で区別する。しかし人数が多いと点呼には時間がかかってしまう。

#### ② マークシート

本大学ではマークシートの出席カードもあり、これを配布することもできる。ただしマークシートの読み取り機に 1 枚ずつ通す必要があるなど手間がかかる。

#### ③ SQS+シートフィーダ付スキャナ

久保先生が開発したアンケート生成集計システム。自動給紙が可能なシートフィーダ付スキャナを利用することで集計が容易です。

配布するのが紙なので、不正が起きないとは言えない

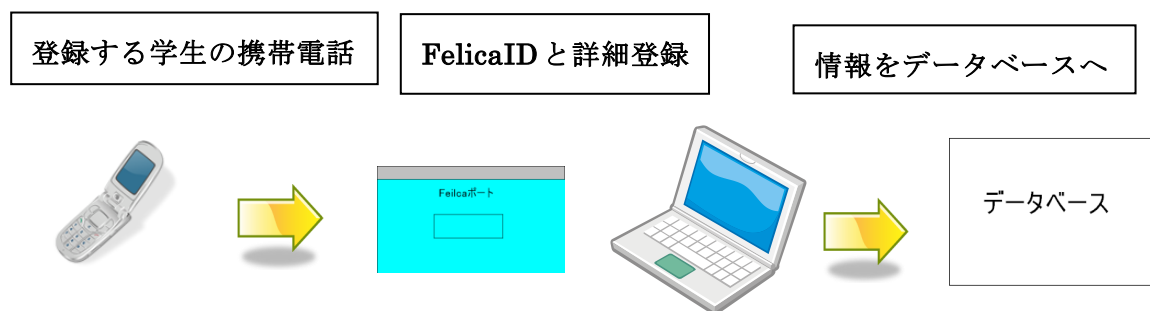
#### ④ QR コード

個人の識別が確実とは言えない

### 3-2-2 Felica を使って出欠情報を登録

まず、出席におけるデータで必要なものとしては、ID と名前と学籍番号、授業などのデータが必要になってきます。（また出席できるかの時間かどうかの情報も必要である。）まずは事前登録のところから始めたいと思います。まずは携帯電話を教室内にある PC につながれた Felica ポートにかざしてもらい feilcaID と自分の情報を登録する。そしてその情報をサーバーに送り情報をデータベースに登録します。（図 3-2-1）してその情報をデータベースからその授業に登録されている人の情報を引き出して出席確認に使う。

図 3-2-1 情報登録



## 4. 設計

### 4-1 目標

今回は生徒情報の登録から出席情報の登録までです。

必要な機能

- ・ 学生登録
- ・ 出席登録

学生：自分のとっている講義

- ・ 集計

先生：自分が教えている講義

事務：全ての先生の講義および全ての学生の出欠

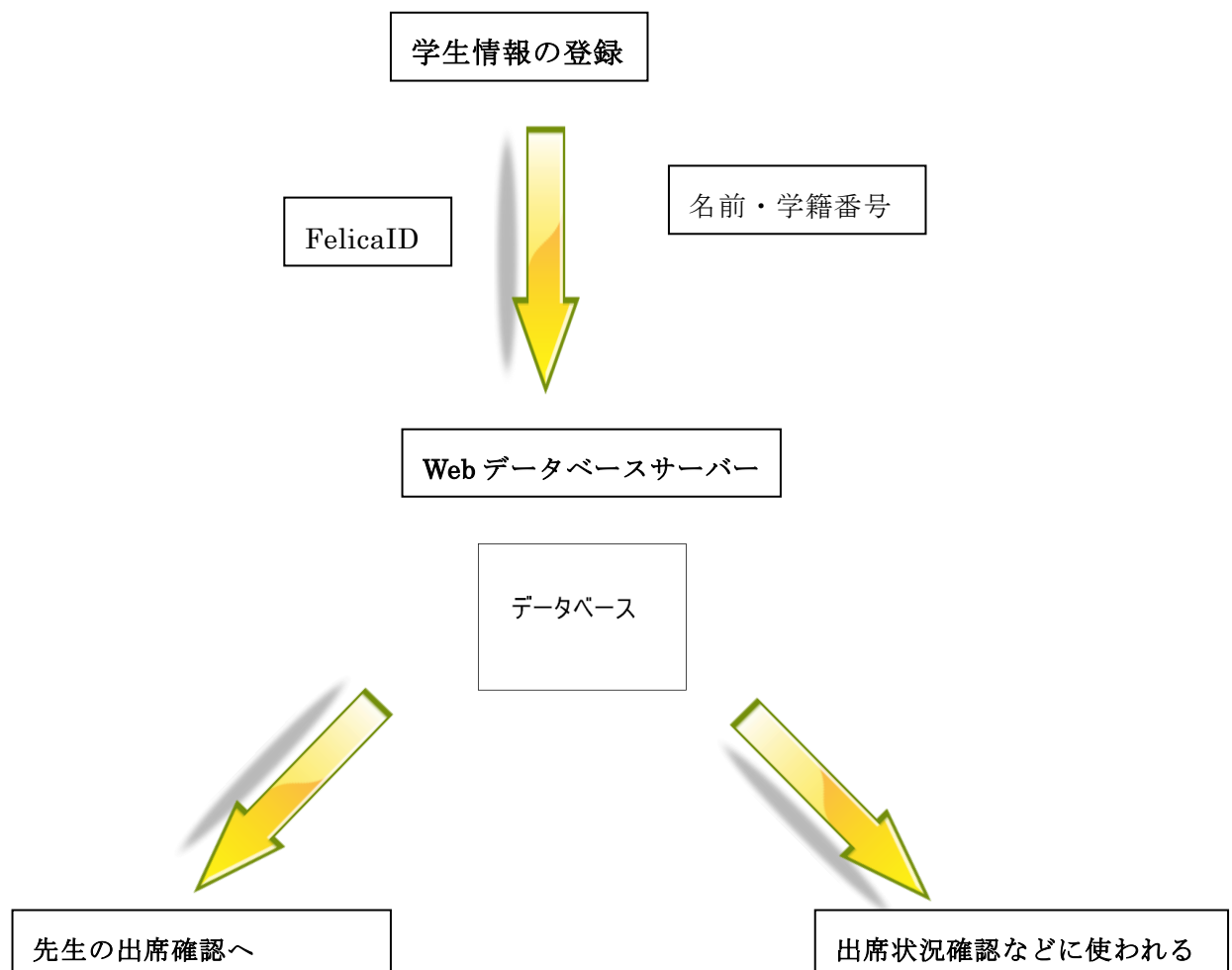


図 4-1 システム構成図



## 4-2 今回のプログラムにおいて必要な要素

今回のプログラムに必要な構成要素は、履修者、先生、事務、携帯電話、データベース、web サーバーが必要となってきます。

### 4-2-1 履修者に必要な要素

- ・ 名前
- ・ 学籍番号
- ・ その人だと第三者でもわかる証明できる ID
- ・ 出席状況

まず、学籍番号が大学から与えられたもので、授業内で重要になっている情報の一つとなっています。しかしそれだけでは第三者からではその人だと学籍番号と名前だけではその人だと判断をすることが出来ない。その人の学籍番号と名前だけ知っていれば誰でもその人になることができってしまうので、学籍番号と名前だけでは判断をつけることが出来ない。しかし、その人専用で誰にもまねをすることが出来ないような証明する ID などがあれば学籍番号と名前と ID の 3 つがそろったとき第三者からでもその人と証明できるものが出来る。また、自分がどれくらい出席していてどれくらい遅刻しているかなどを知りたいことも多々あったりしますが、そういった情報も学生には必要だと思う。

### 4-2-2 先生に必要な要素

- ・ 生徒情報名簿
- ・ 出席状況：遅刻や欠席
- ・ 集計：今まで受けている生徒が遅刻や欠席が何回あるかなど

生徒情報名簿は先生にとって必要なもので、生徒の成績、出席状況などといったことを確認するうえでとても必要なものになっています。

出席状況は、ある生徒が欠席や遅刻または、どうしようもない事情で休んだ場合の欠

席などといったものを調べまたその後の集計で使うために必要なものです。

集計では先ほどあったような欠席、遅刻、欠席届のある欠席などといったものを集計しその生徒の出席がどれほどできちんと授業にでて成績に関係があるかといったものを調べる。

それらがデータとして送信されてくることによって出席などを集計することも出来る。

生徒情報名簿も同様にデータと送られ、送られている情報を確認することができ、その生徒に対しての成績などもつけやすくなる。

#### 4-2-3 事務に必要な要素

- ・生徒情報
- ・出席状況：遅刻か欠席

事務では全生徒の出席状況と先生から送られてくる集計などの作業をする。

生徒の学籍番号や名前、ID などが必要で出席状況などを今までの場合だと先生などから送られてくる出席情報をもとに、データ化して集計するといったことだが、今回の設計においてのことでは出席状況や生徒情報をデータ化して送ることによって集計が簡単になりデータを置いてあるところで集計することにより集計などがもっと早くすることもできる。

#### 4-2-4 携帯電話に必要な要素

- ・携帯固有の ID
- ・個人のものとして証明

携帯電話を使う理由として挙げるのが個人のものとしての証明ができるからである。また、携帯電話に Felica 機能があることが使う理由である。その二つのことによりその人個人のものであったり、Felica 機能を使って ID を取得することにより個人の認識として使うことができ、ID もその携帯電話からしか同じ ID を取得することができないので個人としての認識もでき、第三者から見ても認識として使うこともできる。

#### 4-2-5 データベースに必要な要素

- ・ 履修者の個人データ
- ・ 履修者の出席データ

今まで上げたことなどをこのデータベースに送ることによって生徒情報の登録から出席状況、集計などといったことがデータベース上であることができる。あと多種多様なデータベースを作ることによって必要なデータベースの閲覧から登録までをすることができる。

#### 4-2-6 Web サーバーに必要な要素

- ・ データベースの管理
- ・ 運営

データベースのデータをインターネット上のPCなどからアクセスしてデータを登録するにはWeb サーバーが必要である。

Web サーバーにデータベースを作っておくことによってすべての出席情報や生徒の情報などを管理して置いておくことができる。また、データの管理と運営で、管理するものとして重要なものは、セキュリティ管理です。Web サーバーにはいろいろな人がアクセスするがデータベースの管理部分を行う部分はユーザーを限定してパスワード等を設定することにより、データを安全に保護する必要があります。

### 4-3 機能要素

#### 4-3-1 履修者の登録

履修者の情報（学籍番号・氏名）などを事務や履修者データベースに送信することにより生徒の情報など情報がデータとして送られてくることにより簡単に集計するこ

とができる。

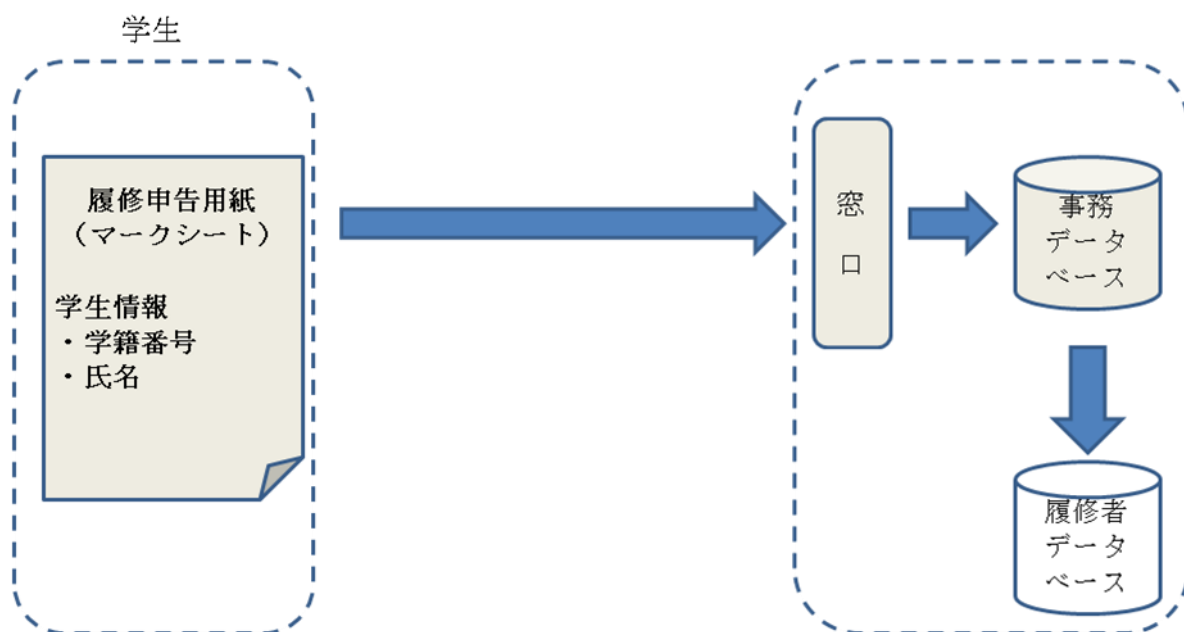


図 4-3-1 履修者登録

#### 4-3-2 履修者の携帯電話を登録

履修者が出席登録に使用する携帯電話を1台を本人の学生情報と対応付けて登録する。これにより出席時に本人確認を行い不正を防止できる。

履修者の携帯電話を Felica ポートがつないである PC で FelicaID を登録する。履修者データベースをその時確認してその情報で合っていたら FelicaID を履修者の FelicaID として登録して履修者データベースが作成される。

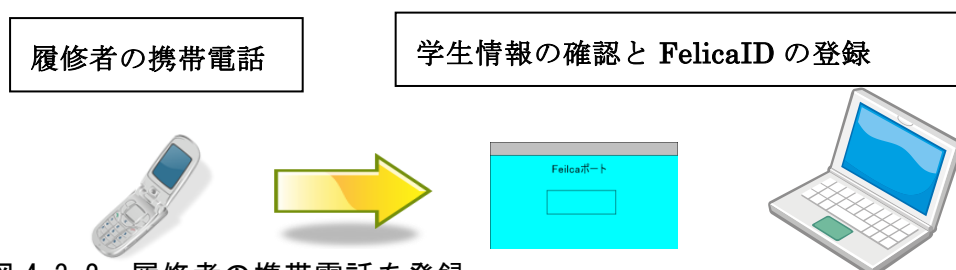


図 4-3-2 履修者の携帯電話を登録

#### 4-3-3 履修者の出席登録

履修者が授業において出席しているときに授業で置いてある Felica ポートにかざすことによって FelicaID を送信して履修者データベースの中から FelicaID で検索して履修者と同じ ID を探してあった場合にはパソコンに表示し、出席時刻を登録して履修者の出席状況が登録される。

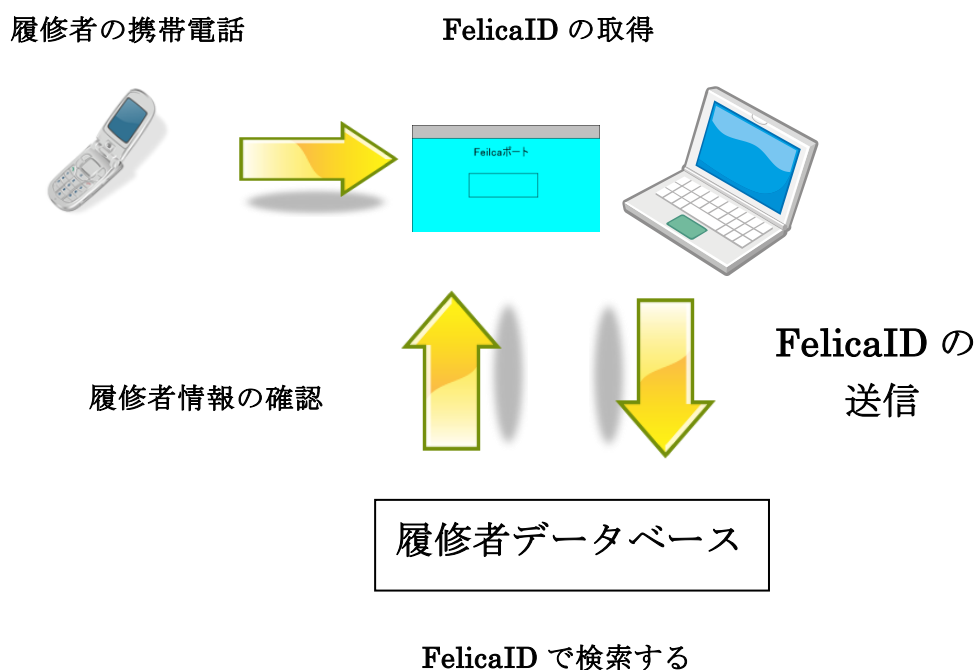


図 4-3-3-1 履修者の確認

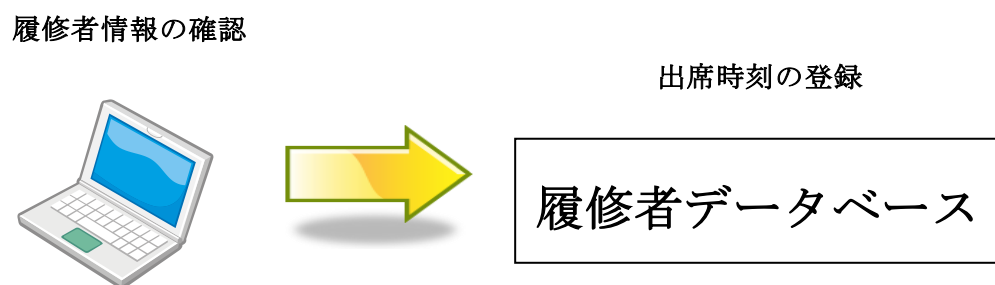


図 4-3-3-1 履修者の出席時刻の登録

#### 4-3-4 先生の出席確認

先生は履修者データベースを担当している講義を指定して確認する。そして履修者の出席状況を確認や集計を行う。

このとき先生からこのデータベースにアクセスするときにはパスワードを入力することによって履修者データベースにアクセスし出席状況の確認と集計が行える。

先生がパスワードでアクセス

出席情報の確認と集計



図 4-3-4 先生の出席確認の流れ

#### 4-3-5 履修者の出席確認

履修者の出席確認は携帯電話を使つての検索となる。出席確認用の端末で携帯電話を Felica ポートなどのカードリーダーにかざすことによってその人個人の認証が行われてその FelicaID で履修している授業の出席情報などが表示される。

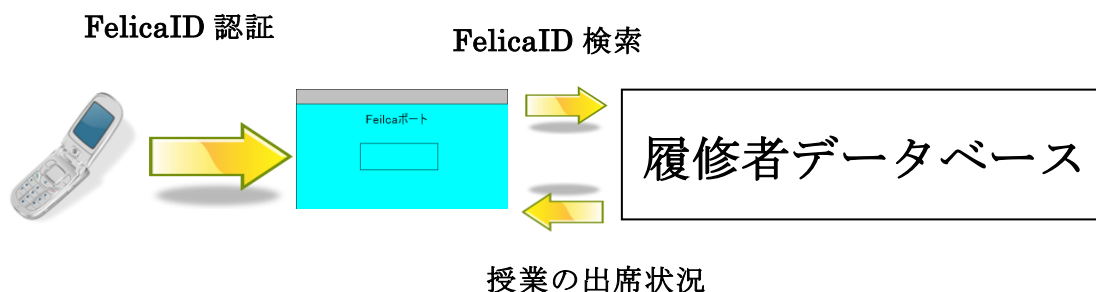


図 4-3-5 出席確認の動き

## 4-4 テーブルの設計

サーバーは現在 PC の中に作って制作しています。(vertrigoseve)

サーバー内部にデータベースを作成しそこにテーブルを作成しデータを登録していきます。

テーブルに必要な情報

### 4-4-1 student テーブル

Student テーブルでは生徒情報の登録をします。

必要なフィールド

- ・名前 varchar
- ・ID char
- ・学籍番号 char

ここに出席情報などを登録していく

### 4-4-2 attend テーブル

Attend テーブルでは生徒の出席状況を登録します。

必要なフィールド

- ・出席 int
- ・学籍番号 char
- ・講義コード char
- ・回数 int
- ・日時 int

ここに出席状況のデータを登録していきます。

## 5. 実装

### 5-1. 実装環境

#### 5-1-1. クライアント環境

オペレーティングシステム : Microsoft Windows XP Home Edition Version 2002 Service Pack 2

CPU : Intel® Core(TM)2 Duo E6850 @3.00GHz

メモリ : 3GB

#### 5-1-2. サーバ環境 VertrigoServ

オペレーティングシステム : Microsoft Windows XP Home Edition Version 2002 Service Pack 2

CPU : Intel® Core(TM)2 Duo E6850 @3.00GHz メモリ : 3GB

Web サーバ : Apache 2.2.8

データベース : MySQL 5.0.51

#### 5-1-3. プログラミング言語

PHP 5.2.4

### 5-2 MYSQL へのテーブル設定

#### 5-2-1 student テーブルの設定



図 5-2-1 student テーブルの画面



## 5-2-2 attend テーブル設定



図 attend テーブルの画面

## 5-2-3 実装システム構成

Send\_id.php で①の FeficaID を取得して POST で送り、②の GET で FelicaID を受け取り、学生情報と対応させて reg\_stu\_id.php に送り、そこから学生情報データベースに③の SQL で登録します。

Id\_kaku.php で④の FelicaID を取得して POST で送り、ID を学生情報データベースから⑤の FelicaID を検索し、同じ物が見つかった場合⑥で id\_kaku.php で表示させる。表示できた場合⑦で POST を使って id\_attend.php に送る。そして、出席状況データベースに⑧SQL で登録する。

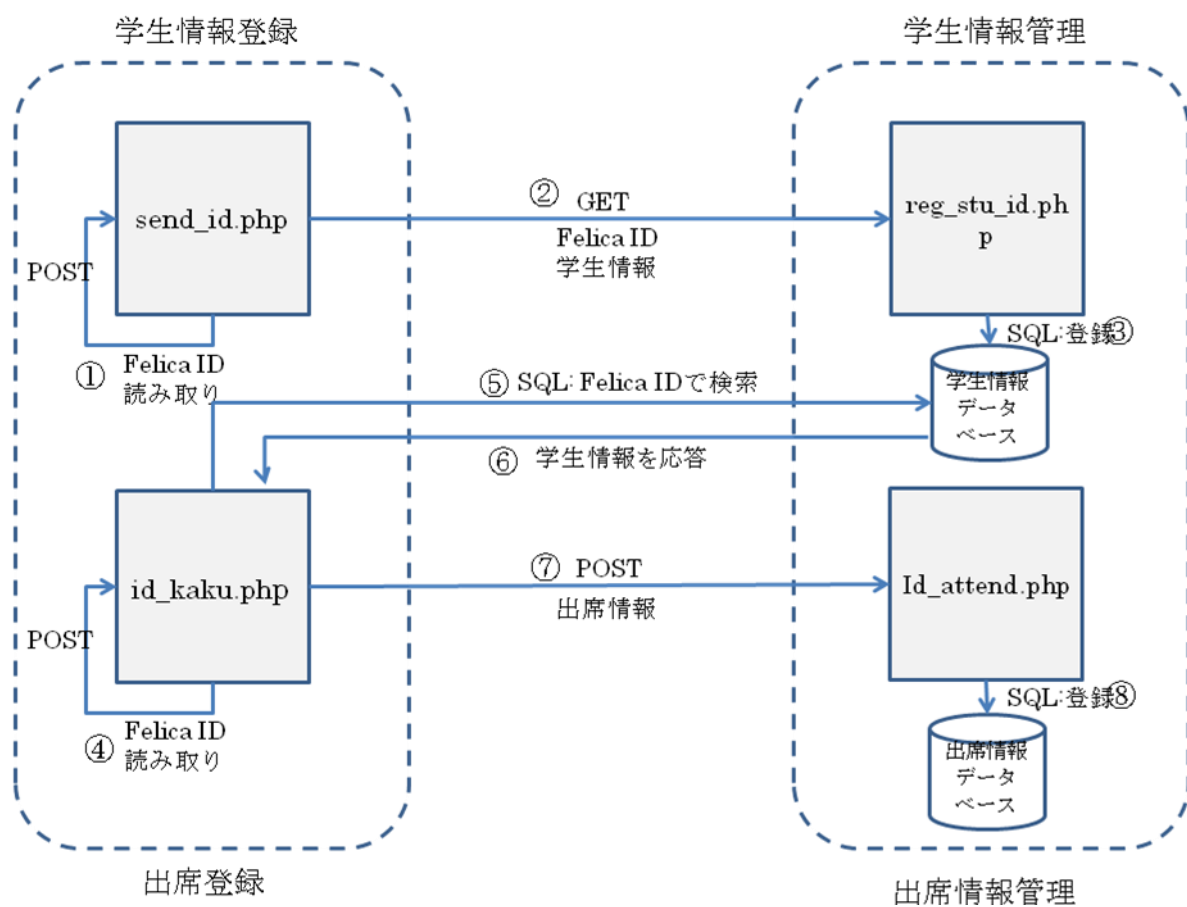


図 5-2-4 実装システムの構成

### 5-3 個人の情報の登録

#### ① FelicaID の取得

FelicaIDを取得するには携帯電話をFelicaポート(図4-2-2)にのせ数上でID取得のプログラムgetmidexeをPHPから実行しその結果を表示させる。

今回 Felica ライブラリといったサイトにあったプログラムを元に作っています。Getmid.exeは携帯電話のFelicaのデータを読み込みFelica機能に付随するIDをそこから取得し表示するといったプログラムになっています。

FelicaIDを取得するためのプログラム(send\_id.php)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```

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

extract($_POST);
$command = '';
if ($kakunin==''){
echo "
<p>カードまたは携帯電話をリーダーにセットしてください。</p>
<p>セットしたら確認ボタンをクリックしてください。</p>
<form method="post" >
<p><input type="submit" name="kakunin" value="確認">
</p>
</form>
";
}

if ($kakunin=="確認"){
    $command = `C:\¥¥test¥¥getmid.exe`;
    echo $command . "<br />";
    if (strstr($command,"failed")){
        echo "読み取れませんでした。<a href=send_id.php> ここを
クリック </a><br />";
        exit;
    }
}

```

```

        echo "ID は、" . $command . "でした。<br />";
        echo "続いて、ID と学生情報を登録します。<br />";
        echo "<a href=¥\"http://127.0.0.1/reg_stu_id.php?id=";
        echo $command;
        echo "¥\"> [クリックしてください] </a><hr>";
        exit;
    }
?>
</body>
</html>

```

## ② 学生情報の登録

①で取得した FelicaID と生徒の情報を対応付けてデータベースに登録する

学生情報の登録のプログラム (reg\_stu\_id)

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
    <head>
        <title>学生情報登録</title>
        <meta http-equiv="Content-Type"
            content="text/html; charset=UTF-8">
    </head>
    <body>
        <?php
            extract($_POST);
            extract($_GET);

            if ($nam<>"") {

```

```

mysql_connect('localhost','root','RX78?ntS');
mysql_select_db('sotu2009');
$sql = "insert into student values(0, '$id', '$nam', '$gak')";
$result = mysql_query($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>";
    exit;
} else {
    echo "正常に登録されました。<br>";
    echo "<a href='\"http://127.0.0.1/send_id.php\"'> クリッ
クしてください </a>";
    exit;
}
}

echo "<p>ID は '$id' です。データを入力してください</p>
<form action='\"reg_stu_id.php\"' method='\"post\"' >
<p>氏 名 : <input type='\"text\"' name='\"nam\"' size='\"40\"'></p>
<p>学籍番号 : <input type='\"text\"' name='\"gak\"' size='\"10\"'>
<input type = '\"hidden\"' name = '\"id\"' value = '$id' >
<p><input type='\"submit\"' value='\"登録\"'>
<input type='\"reset\"' value='\"リセット\"'></p>
</form>";

```

```
?>
```

```
</body>
```

```
</html>
```

③ 生徒が登録した情報を元に出席確認をする。

FelicaID を取得しそれをデータベースにおくり FelicaID を検索する。

一致した場合表示して出席確認をする。

FelicaID 検索用プログラム (id\_kaku.php)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
    <head>
        <title>学生情報確認</title>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    </head>
    <body>
<?php

mb_language("uni");
mb_internal_encoding("utf-8");
mb_http_input("auto");
mb_http_output("utf-8");

extract($_POST);

$lec_code = 56428; //今回は決め打ち、このコードから講義名を検索して
表示できるとよい。
$lec_num = 11; //今回は決め打ち、先生および管理者による入力か、先生
```

の携帯電話をかざすと自動的に入力できるとよい。

```
$command = '';
```

```
echo "<H1>情報倫理論 第" . $lec_num . "回 出席受付</H1></ br>";
```

```
//ここも本来ならば、講義名などは自動で切り替わるほうがよい。
```

```
if ($kakunin==''){
```

```
echo "
```

```
<p>カードまたは携帯電話をリーダーにセットしてください。</p>
```

```
<p>セットしたら確認ボタンをクリックしてください。</p>
```

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

```
<p><input type="submit" name="kakunin" value="確認">
```

```
</p>
```

```
</form>
```

```
";
```

```
} else {
```

```
//外部プログラム実行
```

```
if ($kakunin=="確認"){
```

```
    $command = `C:¥¥test¥¥getmid.exe`;
```

```
    echo $command . "<br />";
```

```
    if (strstr($command,"failed")){
```

```
        echo "読み取れませんでした。<a href=id_kaku.php>
```

```
ここをクリック </a><br />";
```

```
        exit;
```

```
    }
```

```
}
```

```

mysql_connect('localhost','root','RX78?ntS');
mysql_select_db('sotu2009');

$command = rtrim($command);

$sql= "select * from student where id = '$command'";

$result = mysql_query($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 "あなたのデータは以下のとおりです。確認した  
らボタンをクリックしてください。";

            echo "<p>";
            echo "番号: " . $row["renban"];
            echo "<br />";
            echo "ID: " . $row["id"];
            echo "<br />";
            echo "氏名: " . $row["nam"];
            echo "<br />";
            echo "学籍番号: " . $row["gak"];
            echo "</p><hr />";

            echo "<form action =

```



```

¥"http://127.0.0.1/id_attend.php¥" method = ¥"post¥">";
        echo "<input type = ¥"hidden¥" name = ¥"lec_code¥"
value =¥"$_lec_code¥">";
        echo "<input type = ¥"hidden¥" name = ¥"lec_num¥"
value =¥"$_lec_num¥">";
        echo "<input type = ¥"hidden¥" name = ¥"gak¥" value
=¥""";
        echo $row["gak"];
        echo "¥">";
        echo "<p><input type=¥"submit¥" value=¥"確認
¥">";
        echo "</form>";

    }

}

?>
</body>
</html>

```

- ④ 検索された結果から送られてくる情報をデータベースに送信し出席についての情報を送信する

検索された結果からデータベースに出席状況を送信し登録するプログラム。

(id\_attemd.php)

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
    <head>
        <title>出席登録</title>
    </head>

```

```

        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">

        </head>

        <body>

<?php

//サーバ側

mb_language("uni");
mb_internal_encoding("utf-8");
mb_http_input("auto");
mb_http_output("utf-8");

extract($_POST);

$attend_time = time();
$sql = "insert into attend values(0, '$gak', '$lec_code',
        '$lec_num', '$attend_time', 1)";

#echo $sql . "</ br>";

#exit;

mysql_connect('localhost','root','RX78?ntS');
mysql_select_db('sotu2009');

$result = mysql_query($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 "正常に登録されました。<br>";
    echo "<a href='\"http://127.0.0.1/id_kaku.php\"'>出席受付ページ  

    に戻る</a>";
}
?>
</body>
</html>

```

#### 5-4 実行結果

FelicaID の取得と学生情報の登録をする。実験段階のため localhost の send\_id.php にアクセスする。



図 5-4-1 FelicaID 取得画面

図 5-4-1 の画面が表示されたら携帯電話を Felica リーダーにおいて確認ボタンをクリックする。

ID が読み取れた場合は図 5-2-2 のような画面になる。



図 5-4-2ID が取得できた場合

ID が読み取れなかった場合は図 5-4-3 の画面が表示される。



図 5-4-3 できなかった場合

FelicaID が取得できたら図 5-4-2 の画面のクリックしてくださいとなっている部分をクリックしてもらい学生情報の登録 reg\_stu\_id.php に移動する。



#### 図 5-4-4 学生情報の登録

図 5-4-4 の画面で必要な情報を入力し最後に登録を押してもらって学生の情報はデータベースに送信されて登録となります。

次に登録された生徒の情報を FelicaID を使って検索し、生徒情報をデータベースから検索し一致した場合生徒かどうかを表示して授業の出席確認をする。

FelicaID を携帯電話から取得し、データベースで FelicaID を検索して生徒の情報を表示する。

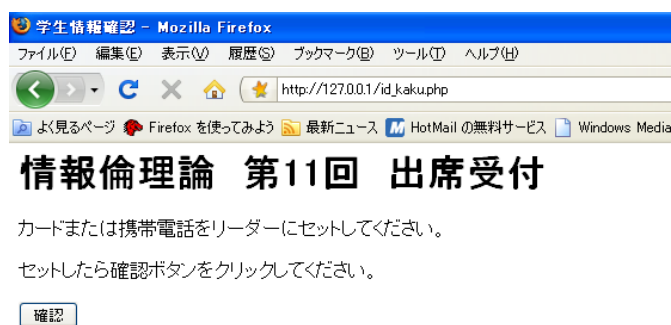


図 5-4-5 FelicaID を取得しデータベースから検索し表示させる。

ここで Felica ポートに携帯電話をかざし、確認ボタンをクリックしてもらう。

そして生徒情報が表示された場合図 5-4-6 のような画面になりもしも FelicaID が取得できなかった場合は図 5-4-7 のような画面になり、FelicaID が検索しても表示されない場合には図 5-4-8 の画面になる。

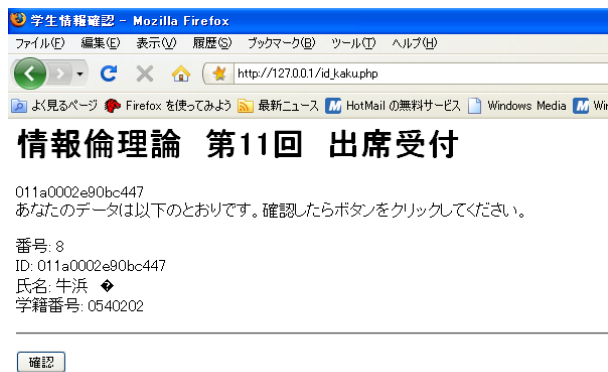


図 5-4-6 生徒情報が取得できた場合



図 5-4-7 FelicaID が取得できなかった場合



図 5-4-8 FelicaID が検索してもデータベースに登録されていない場合

そして図 5-4-6 の画面で自分の情報が確認できた場合そこで下にある確認ボタンをクリックしてもらうと図 5-4-9 の画面に変わることにより出席状況の登録が成功する。

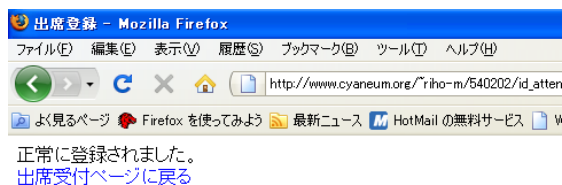


図 5-4-9 出席状況の登録画面

下にある出席受付ページをクリックすると図 5-4-5 の画面に戻り次の人の出席状況の登録に移ることができる。

## 5-5 動作

今回作った生徒情報を登録するプログラムと出席状況を登録するプログラムで両方ともで起きていることは最初に FelicaID を取得するときに一度 ID 取得できないときが多々あるためそういった部分を今後の課題としたい。またそれ以外のデータベースに登録するプログラムでは動作が特に問題なく登録できています。

## 5-6 考察

今回作っていった中での問題点としては最初に FelicaID を取得するときにカードを読み取れないときなどが結構ありました。そこを改良し取得がきちんとできるように今後の課題とします。

また出席状況登録では授業コードや回数などといったところが決めうちになっておりそこを自動で授業コードや回数などを取得し表示させることが出来るよう次の課題としていきます。

現在状態で出席状況の登録までに約 5 秒ほどかかります。しかし、大人数になった場合、100 人だったら  $5 \times 100$  で 500 秒、分に換算して約 8 分くらいで出来ます。しかし、ID が取得できなかった場合も考えると約 10 分くらいで出席状況の登録が出来ます。

ここまでの事からこの状態ではまだまだ完全な実装とまではいけませんでした

が。授業などを限定してやる場合ならば実装可能です。



## 6・まとめと今後の課題

### 6-1 まとめ

本研究では、日常生活で携帯電話を使って何かシステムの構築を目指した。現在では2章でも紹介した札幌大学で行われているシステムがありましたが、そりに近いものになってしまいました。とりあえず現在のものでは生徒情報の登録と出席状況の登録を実装し、データベースに生徒情報を登録すること生徒の出席状況を登録することが出来ました。

### 6-2 今後の課題

今回実装は出来なかった位置情報については、履修コードやデータを取得した端末が置いてある教室 ID などを使って位置情報として大学内で表示するものを現時点では考えている。また出席状況の登録に関しても先生の携帯電話やカードをかざすことによってその授業の一覧が表示されて授業の設定が一気に出来てしまうや先生が自分のしている講義内の生徒出席状況を Web からアクセスすることで閲覧することが出来るものや事務の方に先生の講義や全生徒の出席状況を送信したり、生徒が自分の今受けている講義の出席状況がどうなっているかを閲覧するためのものなどまだ多くの部分が今後の課題として残っています。

## 参考文献

- 【1】 著者 宮沢弦 椎葉宏 片岡俊行 新上幸二 横山隆治 手嶋浩己  
小  
暮祐一 共著 タイトル Mobile2.0 出版社 株式会社インプレスジ  
ャパン
- 【2】 発行人 井芹昌信 タイトル インタージェットマガジン別冊  
web2.0 への道  
出版社 株式会社インプレス R&D
- 【3】 著者 紙谷 歌寿彦 タイトル はじめての人のためのかんたん  
PHP+MySQL 入門  
出版社 秀和システム
- 【4】 著者 田中 ナルミ タイトル MySQL&PHPweb データベース入門  
出版社 ソフトバンクパブリッシング
- 【5】 札幌大学出席確認サービス  
<<http://www.docomo-sys.co.jp/cstudy/cstudy10.html>>
- 【6】 JSTTimeRecorder  
<[http://www.forest.impress.co.jp/article/2006/03/07/jstimerecorder.ht  
ml](http://www.forest.impress.co.jp/article/2006/03/07/jstimerecorder.html)>
- 【7】 Felicalib  
<<http://felicalib.tmurakam.org/>>
- 【8】 Sony Japan felica ホームページ  
<<http://www.sony.co.jp/Products/felica/>>
- 【9】 IC SFCard Fan  
<<http://www014.upp.so-net.ne.jp/SFCardFan/>>

## 謝辞

本研究をするにあたり、担当教諭である渡辺恭人准教授に感謝します。論文の構成から・プログラミング・執筆といった多くのことを御指導いただいたお陰で、自分の実力以上のすばらしい卒業論文を書くことが出来ました。感謝の念でいっぱいです。渡辺ゼミでは、いつも和やかででも的確な指導や出来ないときなどのアドバイスなどしていただきました。モバイルとは何か？といったことから今のモバイル事情や環境の変化などについていろいろと教えていただきました。授業の中で、自分の興味を持つ情報やテーマといったことを教えていただきました。その中から先生と話し合いながら今回の研究が始まりました。この研究が卒業論文として提出できたことは大変うれしい限りです。実装までの間先生には問題が起こるたびそのつどアドバイスをしていただきました。ここまで研究が進めたのは渡辺准教授がいたお陰です。最後に、私の卒論に関わって下さった方全員にもう一度感謝を述べさせて頂き謝辞とさせていただきます。ありがとうございました。