

2011年度 卒業研究

研究題目

「出席状況を自己管理する機構の構築」

指導教員：渡辺 恭人

0840043

齊藤 雷太

目次

1.	背景・目的	3
1-1	背景	3
1-2	目的	3
2.	現状・問題点	4
2-1	現状と問題点	4
2-2	現状の調査方法	4
2-3-1	学生へのアンケート	4
2-3-2	アンケート結果の分析	9
2-4	先生への取材調査	10
2-5	その他アンケートや調査から得られた意見や考察	10
2-6	問題点	11
3.	解決法の検討	12
3-1	アプローチ	12
3-1-1	アンケートから考える	12
3-1-2	学生の関心	12
3-1-3	手段と確実性	12
3-1-4	記録	13
3-2	求められる機能	13
3-3	必要な条件	14
4.	設計と実装	15
4-1	設計	15
4-1-1	システム全体のイメージ	15
4-1-2	登録・閲覧・検索・出欠ページのイメージ	15
4-1-3	データベース	17
4-2	実装	17

4-2-1	授業の登録	17
4-2-2	登録完了ページ	19
4-2-3	登録データの全件を表示	21
4-2-4	指定したデータの出力、閲覧	25
4-2-5	削除	29
5.	評価・考察	32
5-1	動作	32
5-1-1	動作環境	32
5-1-2	動作確認	32
5-2	機能評価	32
5-3	利用者の評価	34
5-4	考察	35
6.	まとめ・今後の課題	36
6-1.	まとめ	36
6-2.	今後の課題	36
	参考資料	36
	謝辞	37

1. 背景・目的

本章では本研究を行うにあたり、理由となった背景と本研究の目的を述べる。

1-1 背景

大学の授業においては出席が単位取得の要件となることは少なくない、しかしながら、学生の中には自分の出席状況がわからずに単位を落とす人がいる。せっかく大学に通っているのに履修した授業の最初の頃だけ出席し、自分の出席状況を知らずに単位を落とす人、欠席数が少なくまだ取り戻せる出席状況にも関わらず諦めて授業に出てこなくなってしまう学生がいる。この様に理由は様々だが欠席し続ける人や諦めて出席しなくなってしまう学生がいることがある。高い授業料を払い学業に励む学生たちの中には、サボってしまう学生もいるだろうし、どうしても授業に出られない学生もいるだろう。しかしそれはもったいないのではないだろうか。そこで自分の出席状況を把握できる手段があればある程度の予定の調整等が可能なのではないだろうか、それをデータ化したものであれば簡潔かつわかりやすく把握できるのではないかと考えた。

1-2 目的

本研究では、学生の授業への出席状況をデータ化し自己管理できるアプリケーションによって出席率の向上、意識改善をできるようにすることを目的とする。

2. 現状・問題点

本章では現状を把握するため調査を行い、調査結果を踏まえ問題点を明確にしている。

2-1 現状

出席状況を確認する手段が担当の先生に直接尋ねるしか方法がない。ある先生に質問したところ授業当日でないと正確には授業履修者の出席状況を正確に伝えられない場合があり、授業当日以外では学生が正確に自分の出欠状況を把握することは難しいということがあるようだ。そこで、出席に関する質問を学生、先生の双方にアンケートにより行い、出席主計方法や利用方法、などに関する調査を行い、出席の現状について把握する。

2-2 調査方法

- 学生へのアンケート調査
- 先生への取材調査

2-3-1 学生へのアンケート

実際に自分の授業出席状況をどれほど覚えているのかを知るために「履修した授業の出席状況をどれだけ覚えているかアンケート」というアンケートを実施した。千葉商科大学の学生全学年から 22 人を対象に調査した。

まず「履修した授業の出欠状況をどれほど覚えているか」という問いに次のような回答が得られた。

- 授業は基本全部出ていたので数えていない、
- 2 度程寝過ごしてしまいその日の授業の出席数がうろ覚えになってしまい以降気にしなくなってしまった。
- 休んだ日だけ手帳に「その授業を休んだ」と記述しておいたので覚えていられた。
- 授業で毎回配布される資料の枚数や資料に書かれている日付で出席した日を確認する。おおまかにこれで確認していた。

- 寝過ごしてしまうことや課題が終わらずに遅刻や欠席を多くしたと思う。数はおおまかに数えていたから欠席数が多くて単位を落としたりもした。
- 大学1年のときは数えていました。その後はほとんど覚えていません。確認できるものとかは特になし。
- 出欠席はほとんど把握していないけど、ほとんど単位は取れているのだから少なかったのだと思う。
- 授業で配布されるプリントの日付で確認し、取ったノートにその日の日付を書いているのでそれでだいたいを把握していた。
- 多少は覚えている、記録などはつけていない。

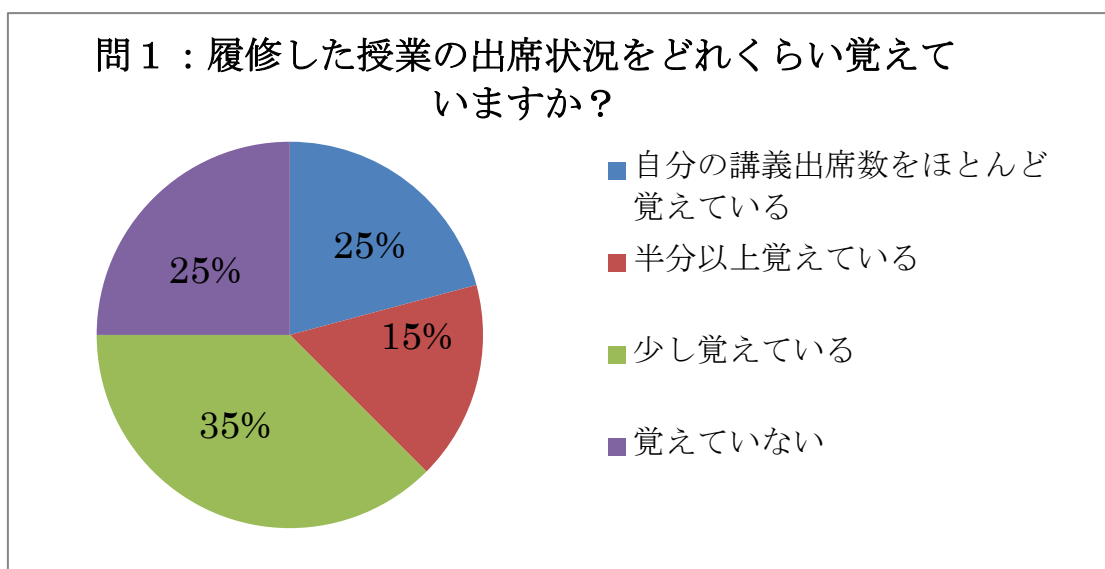


図 2-1. 出欠状況に自己把握の結果

この結果から授業に必ず出席している学生やノートや手帳等にメモを取り自分の出席状況を記録している人は出席率も高く、逆に自分の出席状況を何かしらの媒体に記録を取っていない人は欠席数や遅刻が多く授業に対して消極的な人が多いことや記録をつけない学生が多いことがわかった。また、覚えていないと回答した人の中には無遅刻無欠席の方もいる。半分以上覚えていると回答した人やほとんど覚えていると回答した人の中には覚えていなかった授業の単位を落としているという人が居た。

次に「問：2 出席状況を確認したいと思ったことがあるか、ないか」と質問した結果、以下のようになった。

- 確認したいと思ったことがある (YES) 14 票 64%
- 確認したいと思ったことはない (NO) 8 票 36%

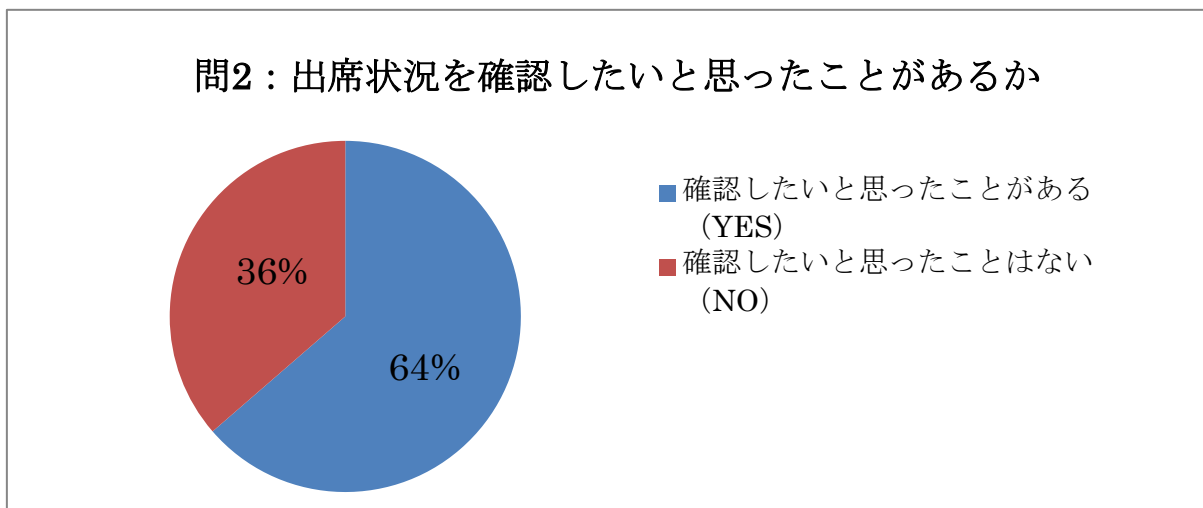


図 2-2. アンケート結果 2

この結果から授業の出席状況を確認したいと思っている学生が多いことがわかる。確認したいと答えた回答者の中には「授業の無い日に確認したくなる」といった人が多くいた。これはいつでも自分の出席状況が確認したいと考えられる。

次に「問3：思ったことがある (YES)」と答えた人に「確認したいと思った理由を簡単に回答してください」と質問した結果以下のような回答が得られた。また、似た回答をまとめグラフにした。

- 単位が取れているか確認したいため
- 何回出席したか知りたいので
- まだ単位が取れるか知りたいため
- 自分が欠席した回数が知りたいから
- 出席になっているか不安だから

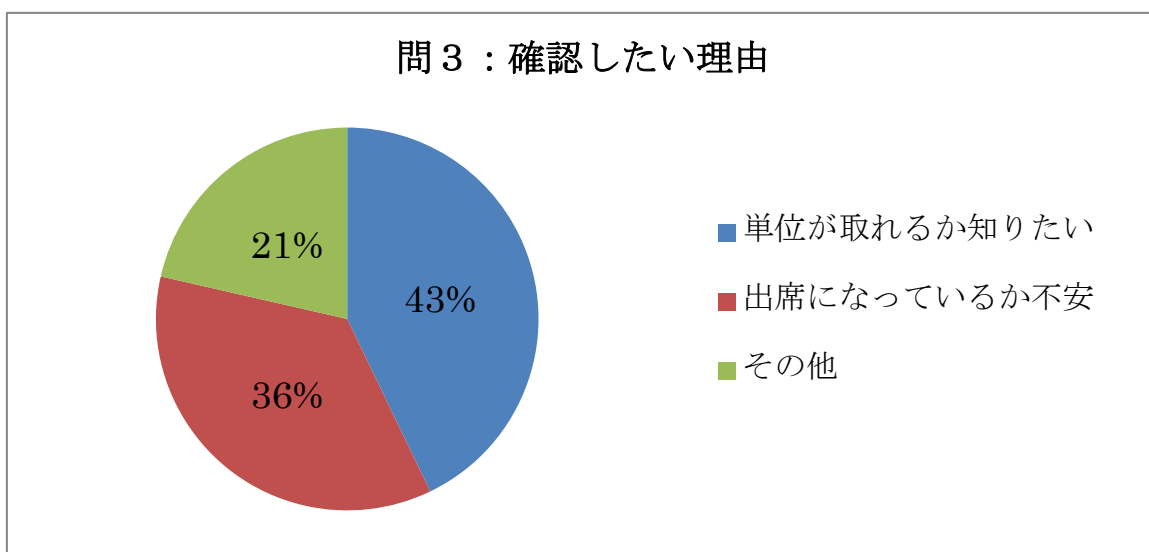


図 2-3. アンケート結果 3

次に「問4：思ったことがある（YES）」と答えた人に「どうやって確認したか簡単に御答えください」と質問した結果、以下のような回答が得られた。

- 授業当日に先生に直接聞く。
- 授業終了後に先生へ確認しに行く。
- 別授業にて先生が同じ人だったので聞きに行った。
- 先生が授業のある日に学内で探して確認しようとした。

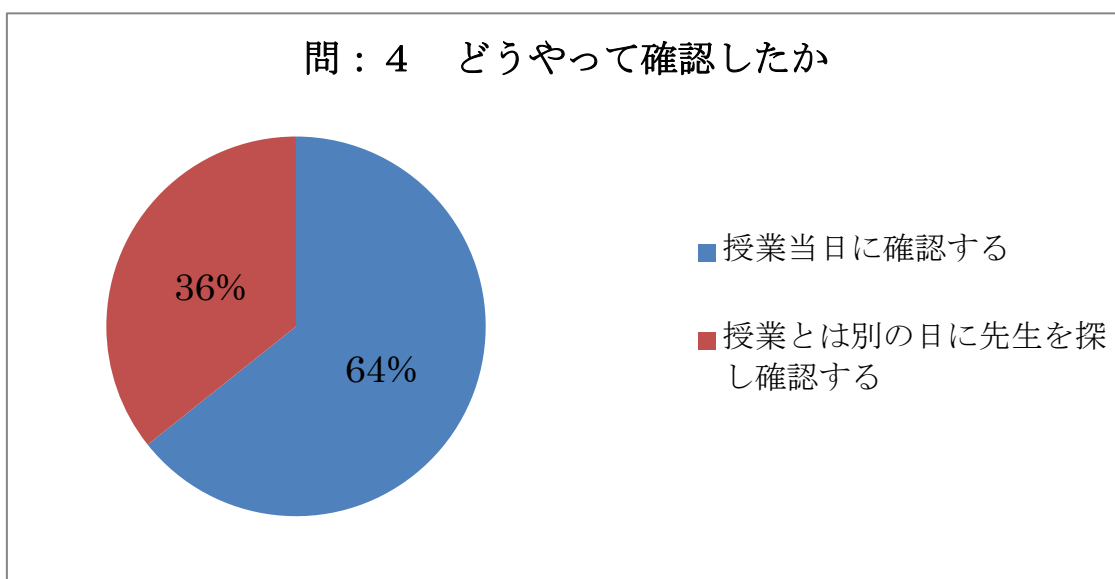


図 2-4. アンケート結果 4

似ている回答をまとめたところ、確認したい授業当日に先生へ聞きに行く場合と先生を探し直接訪ねる場合の2つに分かれていることがわかった。

次に「問5：思ったことはない (NO)」と答えた人に「何故確認しなくても良いと思ったのか簡単に御答えてください」と質問した結果、以下のような回答が得られた。

- 自分の所持している授業の配布物で出席状況がだいたいわかるから。
- ノートを授業の時に必ず取り、ノートを取らない授業の場合は手帳に記述しているから。
- 休んだ日を手帳に記述するので欠席した日はわかる。
- 全部出席しているのだから確認する必要はない。

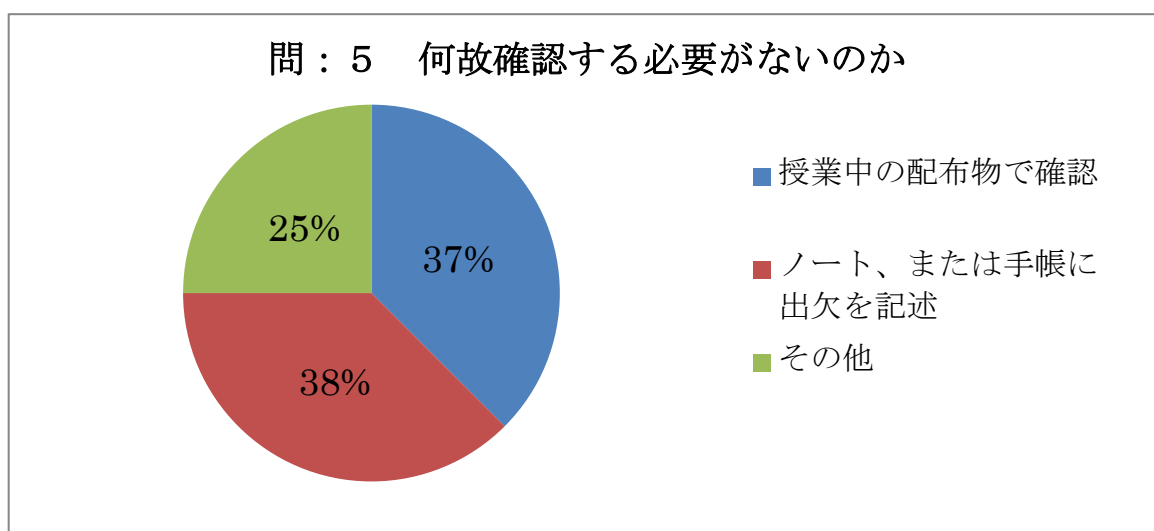


図 2-5. アンケート結果 5

この結果からわかる事は、確認する必要がないと考えている人は授業中に先生から配布される参考資料についている日付や、授業中に取ったノート、または手帳等に自分の出欠を記述する等によって自己管理を行っていたためである。

次に「問6：自分の出席状況を正確にいつでも確認できるとしたらどうですか?」という質問に「助かる」、「助からない」、「どうでもいい」、の3つの中から回答してもらった結果以下の様な割合となった。

■ 助かる	14人	63%
■ 助からない	3人	14%
■ どうでもいい	5人	23%

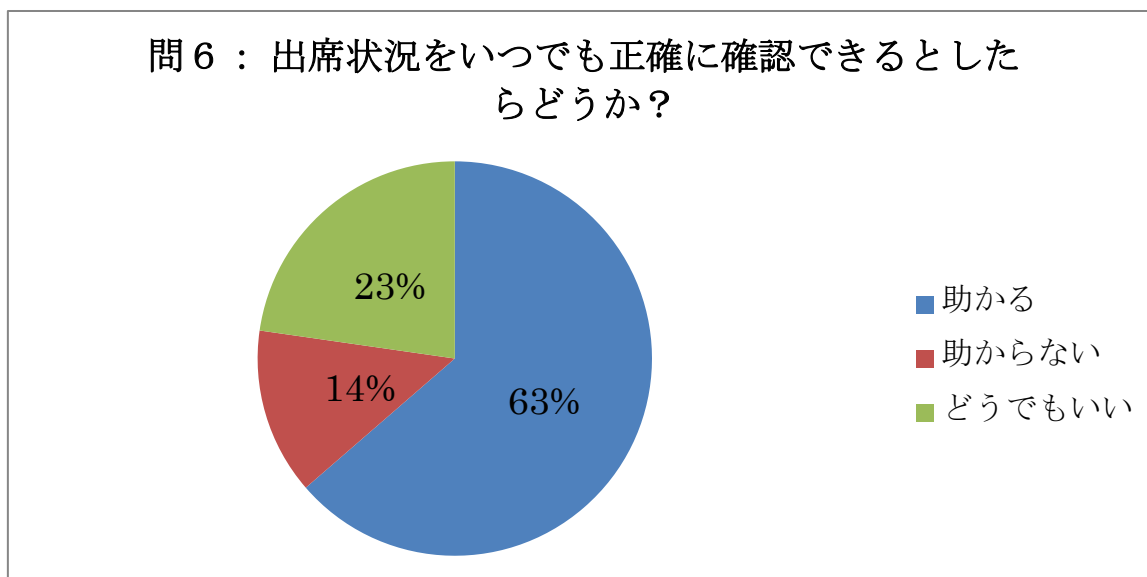


図 2-6. アンケート結果 6

この結果から出席状況をいつでも確認できることに需要があり、自分の出席状況を気にしている人が多いということがわかる。

2-3-2 アンケート結果の分析

授業の出欠状況を確認したいと考えている生徒は多く、主に確認したい授業とは別の日に確認を取ろうとする生徒がいる。確認する理由は履修している授業の単位取得に関する出欠状況の確認をするためである。これは履修者が授業に対して積極性を持っていることを示していると考えられる。確認する方法は大きく分けると授業当日に先生に聞きに行くパターンと授業担当の先生を探し直接尋ねるパターンの2種類がある。先生に確認を取りに行かない学生の多くは手帳やノートに自分の出欠状況を記しておくことにより自己管理をしていた。

確認を取るのは授業当日以外では難しく先生側も必ず対応できるとは限らない、そ

もそも非常勤の教員の方の講義の場合当日以外での確認方法はメール等の通信手段を除くと確認手段が無いものと考えられる。

2-4 教員への取材調査

大学で授業を行っている先生の方々へ取材調査を行った結果。先生によってさまざまな話を伺うことができた。政策情報学で必修である自然言語の授業を担当する先生は、出席は非常に重要な採点のポイントであり、出席数が少ない学生には授業の際に警告を出すなど行っていた。

先生とは異なるが、ウェルネスという政策情報学部にあるスポーツの授業でも出席は厳しく扱われており学期ごとに一定数以上の欠席をした場合その学期ではウェルネスが受けられなくなり、ウェルネスの予約ページでログインが行えない等の規制もある。

一部の先生は課題提出に重点を置いており、出欠に関してはあまり関心がなかった。出席をしていなければ課題の内容がわからないので関心がゼロというわけではないが自然言語やウェルネスといった出席を重要視している授業よりは出席の扱いが軽いことが取材によりわかった。

2-5 その他アンケートや調査から得られた意見や考察

先生側から見ると課題提出を主に採点の基準としている方が多いため出席に対しての関心や評価は低いという意見が多くある。一方で、出席カード等による管理を行う講義や語学や政策情報学だとウェルネスと呼ばれる授業は出席による評価の高い授業も存在する。

学生側から見ると課題はもちろんだが自分の出席が芳しくない場合、単位を既に落としているのではないかという不安や精神的問題により後半の授業に急に出席しなくなるケースが存在した。

2-6 問題点

アンケート結果や取材を元に問題点を挙げる。

1-1. 出席を知りたいが明確に知ることができない

自分の出席状況を知りたいが知るための手段が少なく、明確にわかる場合は少ない。

1-2. 先生側の対応

教員によって出席管理の仕方はまちまちであるため、先生によっては尋ねてもその場ではわからない場合がある。

2. 確実性

出席状況を確認するには講義担当の先生への接触が必要である。出席状況を一番確実に確認する方法としては講義担当の先生へ直接聴きに行く事が一番であるが、大学に常に先生が居るとは限らない、

3. 先生ごとの出席の見方

先生によっては課題さえ出せていれば出席は最低限出ているだけでも良いという意見もあるなか、語学の授業等の一部の授業を担当する先生は出席にも重点を置いている。先生ごとに出席への見方の違いにも問題があると考えられる。

3. 解決法の検討

本章では2章の調査結果を踏まえ、問題点に対しての解決法について述べる。

3-1 アプローチ

2章の問題点、アンケート結果を踏まえて解決方法を述べていく。

3-1-1 アンケートから考える

出席状況を把握できていないことにより単位を落とす人や、たまたま欠席してしまったために課題等の重要な授業内容を聞き逃して課題提出ができなくて単位落としてしまう人もいる。では逆に出席状況を把握できていた人たちはどうやっていたのか、事前アンケートを元に考える。

3-1-2 学生の関心

履修授業のうち半分以上の出席状況を記憶している人がいたが、覚えていない授業に関しては単位を落としている場合があるということがあった。これは記憶できていたとしてもそれは完璧ではなく。覚えていなかった部分を何らかの方法でフォローする必要がある。確認したいと考えている人の半分以上は確認したいと考えており、これは授業に対して積極性があることがわかる。確認したいと考えている生徒の多くは「単位修得のため」に確認する人と「出欠状況」を確認する人の2種類のタイプに分けることができたが、出席意欲があるということは授業への関心が高いことを示している。

3-1-3 手段と確実性

確認を取ることで信憑性の一番高い方法は講義担当の先生に自分の出席状況を直接訪ねることであるが、授業当日以外では先生の方も色々と忙しく動き回っているため出会うのが非常に難しい、先生の中には常勤講師で研究室を持っているため研究室を訪ねる方法もあるが、いつでも居るわけではなく確実性は高くない、講義担当の先生が全員常勤講師というわけではなく、非常勤講師の先生もいる。非常勤講師は研究室を持っておらず基本的に授業当日以外では学内にいることがないため講義当日以外で出会うことは無いといえる。先生を探して直接訪ねる労力を惜しみメール等の通信手段で確認しようとする人もいた。メールで確認するにも直接聞くことに比べすぐに返答が返ってくるわけではないため即時性に欠ける。またメールで確認しようとする人

の中には「聞く内容が不誠実ではないか」というふうを考える人もおり、メールで確認を考えるも実行できなかつた人もいた。確実性が一番高いが常に確認できるわけではなく、手段も限られているこの確認方法は難しいといえる。

3-1-4 記録

3-1-3項で述べたとおり確認が難しい場合は存在する。そこで先生へ確認する必要がないと答えた人が何故確認する必要がないのかという所に着目してみた。確認する必要がないと答えた人の7割以上の人は出欠を自分で確認できていた。講義によっては毎回資料としてプリントを配布する授業もあるため、配布物のプリントの日付で確認し、抜けているプリントの日付で出欠を確認している人がいた。また、授業中に取ったノートに毎回日付を記入しノートを見れば自分の出欠状況を確認する人がいた。授業を欠席した日のみ手帳に記入して自己管理している人もいた。これらからわかることは自分で何らかの媒体に出欠状況を簡単に記しておきそれを確認することによって自分の出欠状況を知ることができるようにしているということである。手元で確認できるこの方法は便利であり即時性も高く、自己記入による管理法ではあるが確実性も高い。そこで、オンラインページで自分の出欠状況を簡単に記録できるページができたかどうか、現在ほとんどの人が携帯電話やパソコンを所持している。暇な時間や、寝る前等の少しの時間に記録し確認もできるなら時間的な負担も少ないと考えられる。

3-2 求められる機能

3-1節で述べた負担の少ない情報端末を利用した出席の自己管理を行うシステムを作るためには、利用者の授業のある毎週行われる授業の曜日と時間の情報を記録し、その情報を元に毎日の出欠を授業ごとに記録し一覧を作成するという処理の流れが想定される。

本研究では上記の想定される流れを基に、役割別に3つの機能で構成されたシステムを構築する事にした。具体的には、利用者の授業情報を記録する機能、次に授業ごとに出欠を記録し授業の出席状況を分析する機能、最後に分析した結果と出欠数を一覧にして表示を行う機能である。

3-3 必要条件

- ① 閲覧や入力する際の端末を問わないものとして WEB ページである必要がある。
- ② 利用者ごとのデータをまとめるためにアカウントを設ける。
- ③ 利用者が授業を登録、閲覧ができる環境を作る。
- ④ 登録されたデータの編集・削除等の操作は、利用者あるいは管理者のみができるようにする。
- ⑤ アカウント内で登録されている授業情報を利用者が閲覧、検索ができるようにする。
- ⑥ 登録されている情報に出席欠席の情報を登録できるようにする。
- ⑦ 利用者が自分の履修していない授業の情報を知る場として交流の場が必要である。
- ⑧ 交流の場として授業情報を公開、非公開にできる機能、掲示板の様な必要である。

4. 設計・実装

本章では、これまでの議論を踏まえた上での解決法や必要とされる条件から出欠を自己管理するための機能の設計・実装を行い、実装について記述する。

4-1. 設計

機能実装に向けて全体のイメージや構成を記述していく。

4-1-1. システム全体のイメージ

- ① アカウントを設け、アカウント情報をデータベースでまとめる。
- ② データベースの編集・削除等の操作は、管理者のみができるようにする。
- ③ アカウント内で登録されている情報の閲覧、検索ができるようにする。
- ④ 登録されている授業情報に出席欠席の情報を登録できるようにする。
- ⑤ 公開、非公開機能をつけ、他者の履修授業の情報を閲覧できるようにする。

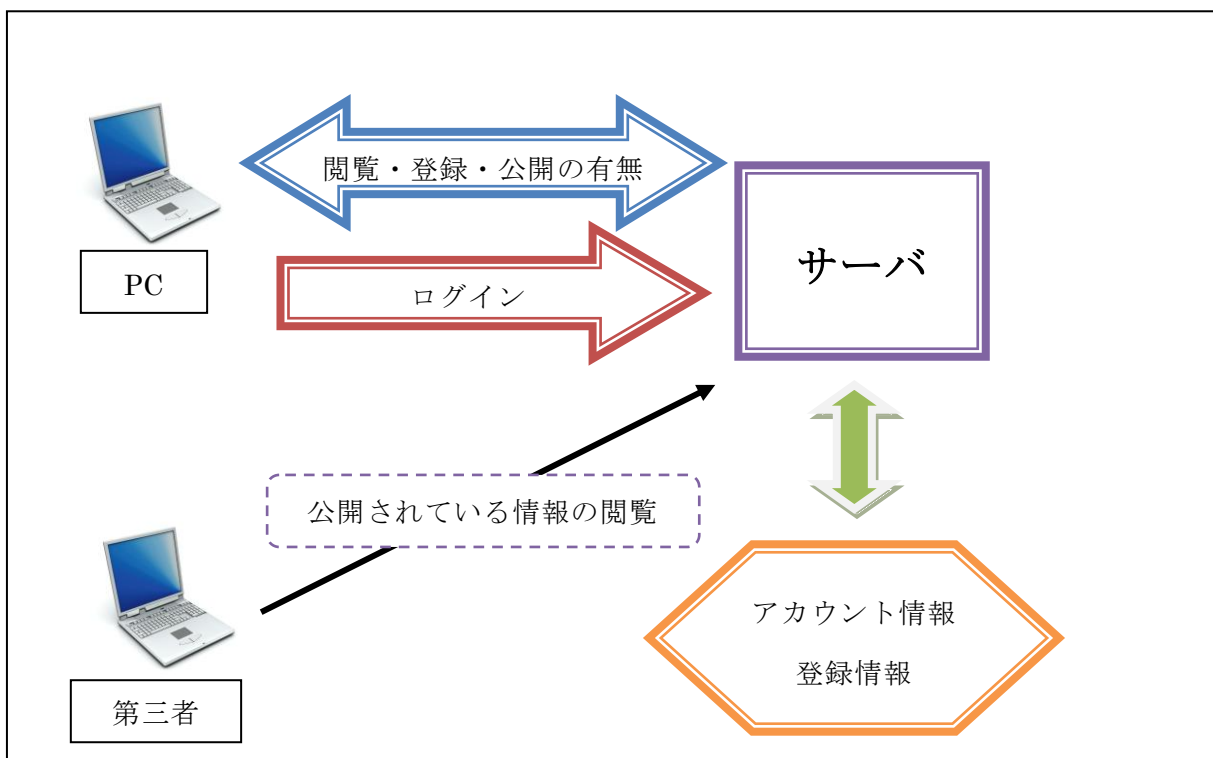


図 4-1-1. システムのイメージ

4-1-2. 登録・閲覧・検索・出欠ページのイメージ

トップページ(top.html)にアクセスするとログイン画面(login.php)が表示される、ログインするとアカウント情報から自分のメインページへ飛ぶ、メインページ(main.html)からは授業情報を新規に登録するページ(entry.html)、登録した授業を全

件表示するページ(wat.php)、条件を指定して授業を検索するページ(ken.php)、登録した授業を曜日、時間ごとに表示するページ(ichi.php)へ飛べるようにする。登録した授業情報は検索ページか全件表示ページから削除ができるようにする。登録情報の公開、非公開ができるようにする。

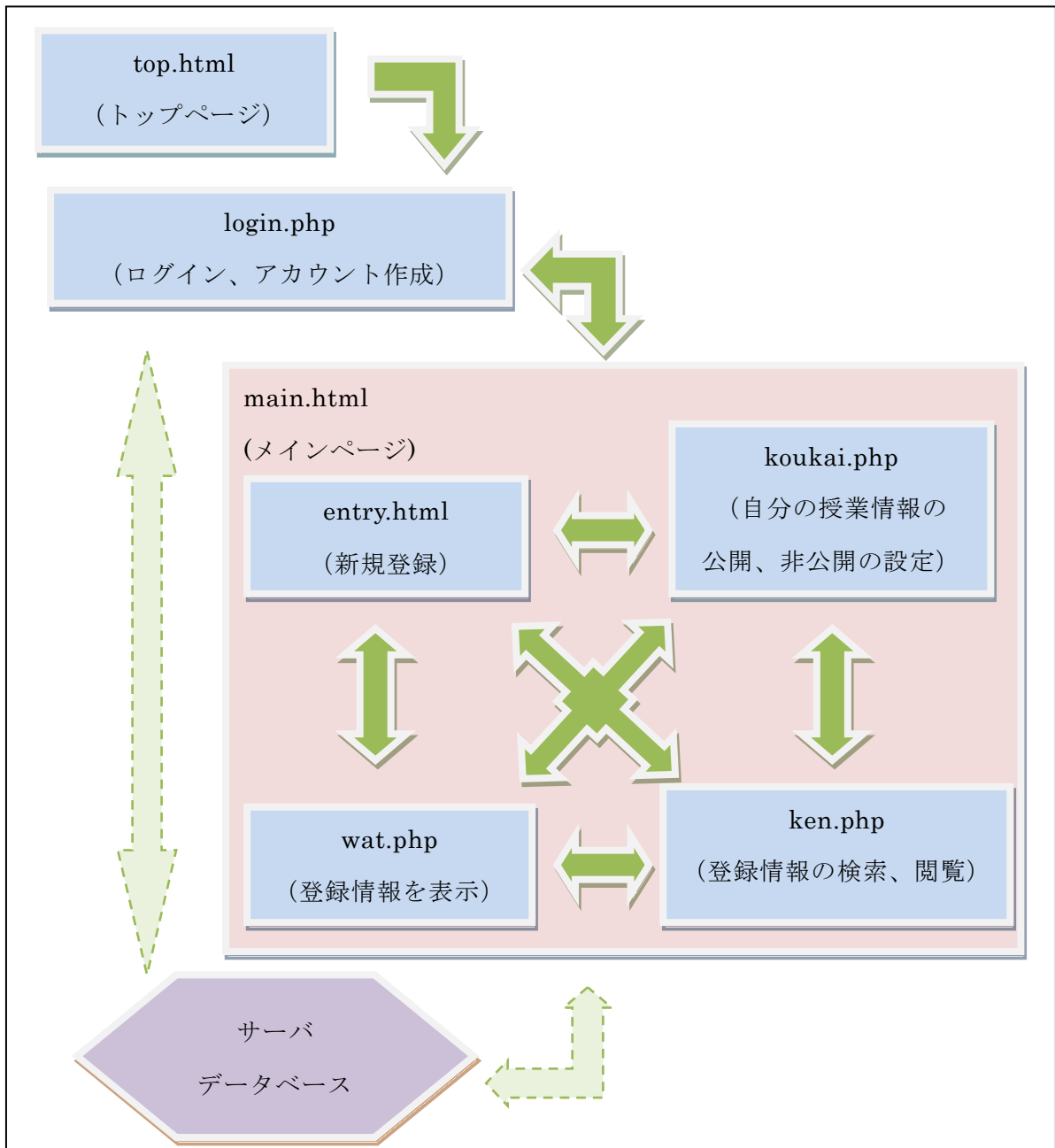


図 4-1-2. 実装構成図

4-1-3. データベース

実装を想定して情報を入力し格納しておくテーブルを Mysql データベース上に作成しておく。登録した情報に番号を振って重複を回避するために renban を作成しておく、データ型は数値を入れるため int 型にしておく。授業の曜日、時限は数が一定数しかないので数値として保存するため int 型にしておく。授業名、備考その他は授業の名前や授業の内容などについて簡単に文字を入力できるようにするため varchar にする。

内容	フィールド名	データ型
重複回避	renban	Int
授業名	name	varchar(40)
曜日	youbi	Int
時限	jikan	Int
備考その他	yotei	varchar(40)

図 4-1-3. テーブル (syuketu4) の設計

4-2. 各機能の実装

設計を元の実装した機能の解説を述べていく。

4-2-1. 授業の登録

授業登録を行う画面である予定入力ページ(図 4-2-1)にて登録したい授業名と曜日、授業のある時間を入力して登録をクリックすると登録完了ページ(図 4-2-1)へ入力した情報が送信される。リセットをクリックすると現在入力している情報を初期化することができる。

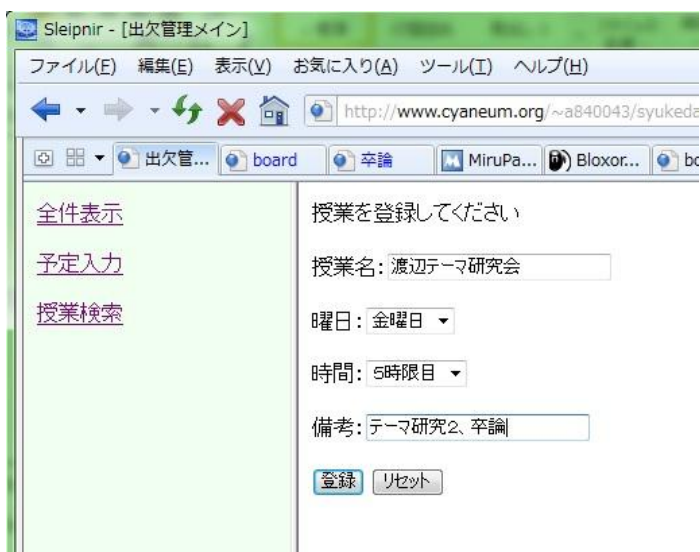


図 4-2-1. 予定入力ページ(syuketu_entry.html)

予定入力（図 4-2-1）のページは表 4-2-1(1)によって処理がされている。曜日や時間ごとに<option value=*>の値が設定されており、見た目は曜日や時間の名称が表示されるがソース内では数値による登録処理が行われている。授業名は input type=text によりテキスト入力ができるようになっている。size=" 30" により全角 15 文字まで入力できるようになっている。この入力された情報は登録完了ページ(図 4-2-2)へと method=" post" によって送られる。

表 4-2-1(1). syuketu_entry.html

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="ja">
<head>
    <title>予定入力</title>
    <meta http-equiv="Content-Type" content="text.html;
charset=Shift_JIS">
</head>
<body>

<p>授業を登録してください</p>

```

```
<form action="syuketu_entry.php" method="post" >
<p>授業名 : <input type="text" name="name" size="30"></p>
<p>曜日 : <select name="youbi">
    <option value=1>月曜日</option>
    <option value=2>火曜日</option>
    <option value=3>水曜日</option>
    <option value=4>木曜日</option>
    <option value=5>金曜日</option>
    <option value=6>土曜日</option>
</select> <br>
<p>時間 : <select name="jikan">
    <option value=1>1 時限目</option>
    <option value=2>2 時限目</option>
    <option value=3>3 時限目</option>
    <option value=4>4 時限目</option>
    <option value=5>5 時限目</option>
    <option value=6>6 時限目</option>
</select> <br>
<p>備考 : <input type="text" name="yotei" size="30"></p>

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

4-2-2. 登録完了ページ

予定入力ページで入力した情報を登録完了ページ(図 4-2-1)で受け取り、入力された情報をサーバに送る。ページである。「登録ページに戻る」をクリックすることにより予定入力ページ(図 4-2-1)に戻ることができる。



図 4-2-2. 登録完了ページ (syuketu_entry.php)

予定入力ページ(図 4-2-1)にて入力された内容、“renban” “name” ” youbi” “jikan” ” yotei” の入力情報はMysqlデータベース上に作られたテーブル内のフィールド名に対応している。予定入力ページで入力された情報は method=post で登録完了ページへと送信されている。この情報を変数\$_POST で受け取り、mysql データベース上に作ったテーブルへと送信される。なお、ここで初めてでる renban と呼ばれるフィールド名は情報の重複を避けるために番号を振っているものである。

表 4-2-2(1). (syuketu_entry.php)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
  <head>
    <title>timetable</title>
    <meta http-equiv="Content-Type"
    content="text/html; charset=Shift_JIS">
  </head>
  <body>
<?php
extract($_POST);
mysql_connect('localhost','root','*****') or die(mysql_error());
```

```

mysql_select_db('lesson_a840043');

$sql = "insert into syuketu4 values('$renban','$name', '$youbi', '$jikan',
'$yotei', '$syuketu')";
$result = mysql_query($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=syuketu_entry.html>登録ページに戻る</a>";
}
?>

</body>

</html>

```

4-2-3. 登録データの全件を表示

登録ページ（図 4-2-1）で登録した情報をサーバからすべて出力し、閲覧することができる。ページ内にある[このレコードを削除する]をクリックすると登録されている情報を1件削除することができる。

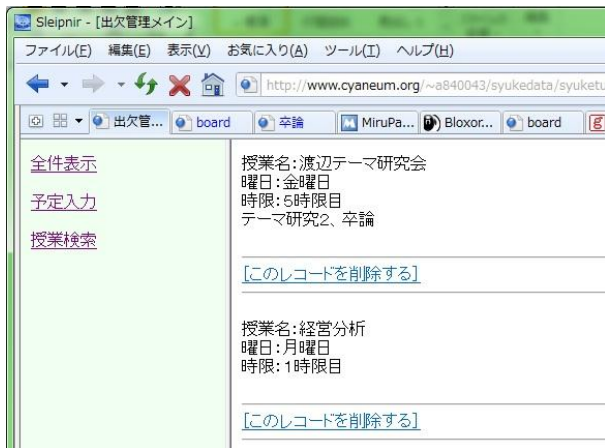


図 4-2-3. 登録情報閲覧画面 (syuketu_watch.php)

入力された授業情報をデータベース上から呼び出し表示する内容が記述されている。予定入力ページ(図 4-2-1) で入力された授業名はテキストとして入力されているためそのまま出力されるようになっている。曜日、時間は select によって曜日ごとに数値型で入力されているため、数値に対応した内容が表示される。[このレコードを削除する]というものはテーブル内のフィールド名を id=というもので指定し、削除ページへと送るものである。

表 4-2-3(1). (syuketu_watch.php)

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
  <head>
    <title>予定一覧</title>
    <meta http-equiv="Content-Type"
    content="text/html; charset=Shift_JIS">
  </head>
  <body>
<?php
mysql_connect('localhost','root','*****') or die(mysql_error());
mysql_select_db('lesson_a840043');

```

```

$sql= "select * from syuketu4";
$result = mysql_query($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 "授業名 : ";
            echo $row["name"];
            echo "<br />";
            echo "曜日 : ";
            if($row["youbi"] == 1) {
                echo "月曜日";
            }
            elseif($row["youbi"] == 2) {
                echo "火曜日";
            }
            elseif($row["youbi"] == 3) {
                echo "水曜日";
            }
            elseif($row["youbi"] == 4) {
                echo "木曜日";
            }
            elseif($row["youbi"] == 5) {
                echo "金曜日";
            }
            elseif($row["youbi"] == 6) {
                echo "土曜日";
            }
        }
    }

```



```

    }
    echo "<br />";
    echo "時限：";
    if($row["jikan"] == 1){
        echo "1 時限目";
    }
    elseif($row["jikan"] == 2){
        echo "2 時限目";
    }
    elseif($row["jikan"] == 3){
        echo "3 時限目";
    }
    elseif($row["jikan"] == 4){
        echo "4 時限目";
    }
    elseif($row["jikan"] == 5){
        echo "5 時限目";
    }
    elseif($row["jikan"] == 6){
        echo "6 時限目";
    }
    echo "<br />";
    echo $row["yotei"];
    echo "<br /> ";

    echo "</p>";
    echo "<hr>";

    echo "<a href=¥"syuketu_delete.php?id="";
    echo $row["renban"];

```

```

        echo "$¥" [このレコードを削除する] </a><hr>";
    }
}
?>
</body>
</html>

```

4-2-4. 指定したデータの出力、閲覧

登録ページ（図 4-2-1）で登録した授業を曜日と時間を指定して検索をクリックすることで曜日と時間の条件に合致した登録済みの授業情報を呼び出し、閲覧することができる。

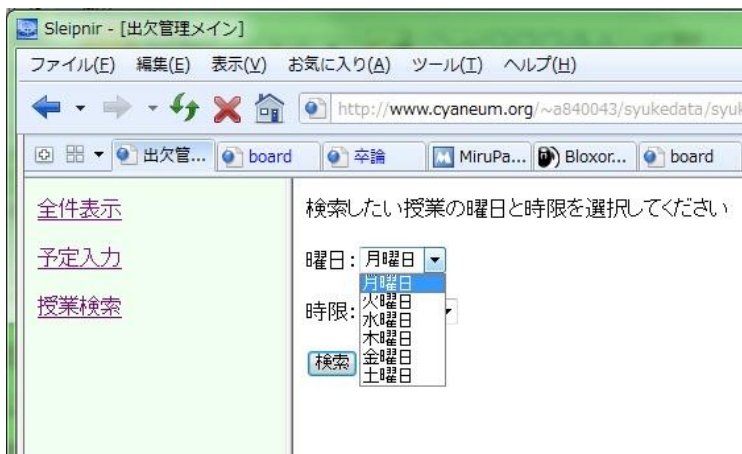


図 4-2-4. 登録情報検索画面 (syuketu_kensak.php)

検索画面では予定入力ページ（図 4-2-1）にて入力された曜日、時間の登録情報を指定し、合致したもののみを表示するというものである。この`\$youbi<>'` `&& \$jikan<>'`という内容により曜日と時間が合致したもののみを表示することができるようになっている。

表 4-2-4(1). syuketu_kensak.php

```

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

```

```

        <title>予定一覧</title>

        <meta http-equiv="Content-Type"
content="text/html; charset=Shift_JIS">
    </head>
    <body>
<?php

extract($_POST);

echo "
<p>検索したい授業の曜日と時限を選択してください</p>
<form action="php echo syuketu_kensak.php ?" method="post" >
<p>曜日 : <select name="youbi">
            <option value=1>月曜日</option>
            <option value=2>火曜日</option>
            <option value=3>水曜日</option>
            <option value=4>木曜日</option>
            <option value=5>金曜日</option>
            <option value=6>土曜日</option>
        </select> <br>
<p>時限 : <select name="jikan">
            <option value=1>1時限目</option>
            <option value=2>2時限目</option>
            <option value=3>3時限目</option>
            <option value=4>4時限目</option>
            <option value=5>5時限目</option>
            <option value=6>6時限目</option>
        </select> <br>
<p><input type="submit" value="検索">
<input type="reset" value="リセット">

```

```

</form>
";

if($youbi<>' ' && $jikan<>' '){
mysql_connect('localhost','root','*****') or die(mysql_error());
mysql_select_db('lesson_a840043');

$sql="select * from syuketu4 where youbi = $youbi and jikan = $jikan";
$result = mysql_query($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 "授業名 : ";
            echo $row["name"];
            echo "<br />";
            echo "曜日 : ";
            if($row["youbi"] == 1) {
                echo "月曜日";
            }
            elseif($row["youbi"] == 2) {
                echo "火曜日";
            }
            elseif($row["youbi"] == 3) {
                echo "水曜日";
            }
        }
    }
}

```

```
elseif($row["youbi"] == 4) {
    echo "木曜日";
}
elseif($row["youbi"] == 5) {
    echo "金曜日";
}
elseif($row["youbi"] == 6) {
    echo "土曜日";
}
echo "<br />";
echo "時限：";
if($row["jikan"] == 1) {
    echo "1 時限目";
}
elseif($row["jikan"] == 2) {
    echo "2 時限目";
}
elseif($row["jikan"] == 3) {
    echo "3 時限目";
}
elseif($row["jikan"] == 4) {
    echo "4 時限目";
}
elseif($row["jikan"] == 5) {
    echo "5 時限目";
}
elseif($row["jikan"] == 6) {
    echo "6 時限目";
}
echo "<br />";
```

```
        echo $row["yotei"];
        echo "<br /> ";

        echo "</p>";
        echo "<hr>";

        echo "<a href=%Y\"syuketu_delete.php?id=";
        echo $row["renban"];
        echo "%Y\"> [このレコードを削除する] </a><hr>";
    }
}
?>
</body>
</html>
```

4-2-5. 削除

検索ページ、全件表示ページにある[このレコードを削除する]により削除情報を送信、削除完了を表示するページである。

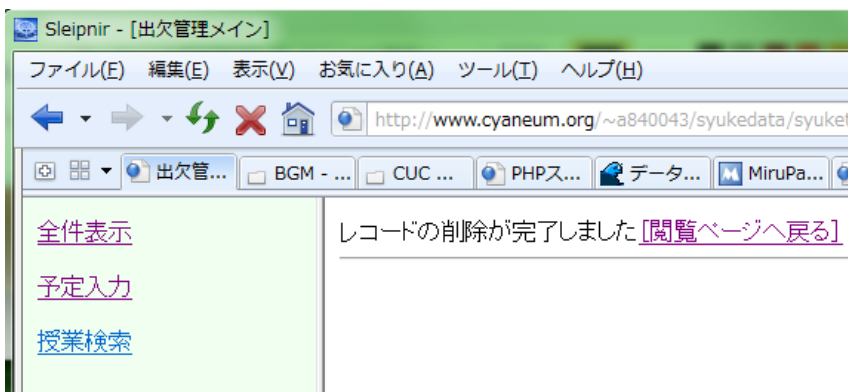


図 4-2-5. 削除完了ページ

4-2-3. と 4-2-4. にある以下の内容について解説していく。

これらは id=によって renban を呼び出し、呼び出した renban に対応した番号のレコードを削除するという内容である。レコード内には登録された授業名や曜日、時間等の登録情報が保存されている。

表 4-2-5. 削除について

```
echo "<a href=¥"syuketu_delete.php?id=";  
echo $row["renban"]];
```

受け取った id=で送られた renban 情報を \$id で受け取り該当する連番を削除する命令を mysql へと送っている。

表 4-2-5(1). syuketu_delete.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html lang="ja">  
  <head>  
    <title>削除</title>  
    <meta http-equiv="Content-Type" content="text/html;  
charset=Shift_JIS">  
  </head>  
  <body>  
<?php  
extract($_GET);  
mysql_connect('localhost','root','20cuc08!');  
mysql_select_db('lesson_a840043');  
  
$sql = "delete from syuketu4 where renban = $id";
```

```
mysql_query($sql);  
echo "レコードの削除が完了しました";  
  
echo "<a href=¥"syuketu_wat4.php?id=";  
echo "¥"> [閲覧ページへ戻る] </a><hr>";  
?>
```


5. 評価・考察

本章では4章で実装された機能の動作確認、評価から考えられる考察を記述する。

5-1. 動作

実装した機能の動作環境、動作確認を行う。

5-1-1. 動作環境

動作確認を行うPCの環境は以下の通りである。

千葉商科大学第1実習室のPCを使用した。

本体：富士通 CELSIUS CLJ8GHT

OS：Windows7 Professional 32bit

CPU：Intel(R) Xeon(R) X3460 2.80GHz 2.79GHz

実装メモリ (RAM)：4.00GB (使用可能メモリ：2.96GB)

5-1-2. 動作確認

利用者である学生はまず、自分の履修している授業を登録するはずである。予定登録ページ (図 5-1-2) から授業名や曜日、時間を指定して登録する。登録した授業がどれくらいあるのかを確認するために全件表示ページ (図 5-1-2-1) をクリックへ飛び、確認をする。または、曜日ごとや時間ごとに自分の登録した授業を閲覧する時は授業検索ページ (図 5-1-2-2) へ飛び、曜日や時間を指定して検索をクリックする。



図 5-1-2. 予定入力

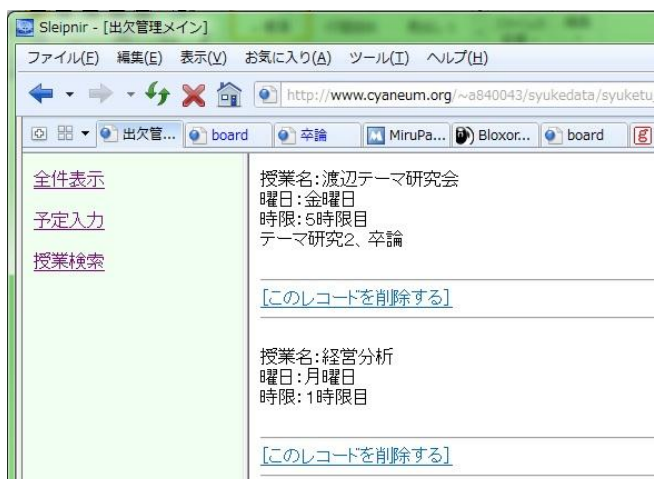


図 5-1-2-1. 全件表示

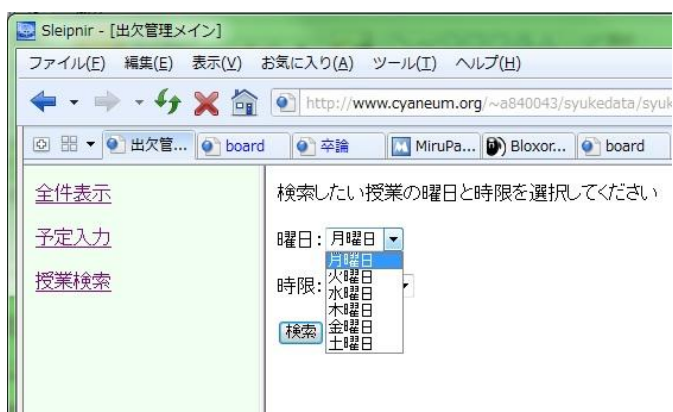


図 5-1-2-2. 授業検索

5-2. 機能評価

表 5-2. 機能評価

設計機能	実装状況
ログインフォーム	×
アカウント管理	×
授業情報の登録	○
登録情報の表示	○
登録情報の検索	○
出欠情報の登録	×
出欠情報の集計と出力	×

授業情報の登録、登録した授業情報の表示、登録した授業情報の検索を実装することができた。ログインフォーム、アカウント管理、出欠情報の入力、出欠情報の集計と出力を実装させることができなかつた。実装できなかつたのは出欠を自己管理するうえで前提となる授業の登録、表示、検索に重点を置いたためログインフォームの設置や処理、アカウント管理や出欠情報の入力、集計、出力を作成するに至ることができなかつた。

5-3. 利用者評価

利用者に設計の目標と目的を伝えたいうえで、実装した機能を利用してもらい、利用者から得た評価を記述する。

- ・ 同じ時間に複数の授業入れても検索で全部候補出してくれるのは良い
- ・ 全件表示も割と見やすい感じだが仮に多く登録した時には逆に見にくいかも
- ・ 全件表示の部分は表に出力できるようになればいいんじゃないかな！
- ・ 授業のある日付入れたいかな何回目の授業でたのかとかがわかりやすいかなーって
- ・ 見た目とかはいいと思うよ
- ・ 俺の感想としては 自分が入力したのがいつの授業なのかとかがわかると使いやすいのかなって思った。
- ・ 左側に別項目でカレンダー表示みたいなのがあると便利かも。ブログパーツでよくあるやつ。
- ・ レコード削除はもうワンクッション削除確認とかがあると個人的には安心だけどこれは個人差かなーって感じです。
- ・ 完成したら便利そうだと思います。
- ・ 出欠管理なら日付とか入れたかったかな
- ・ 登録の仕方が簡単でよかったです
- ・ 単独のアプリとしてならアカウント関連は必要なかったかもね
- ・ 完成していないし、使用できないので評価のしようがない
- ・ 早くアカウント管理作ってね。出席とか授業ごとの備考を書けるようにするなら授業ごとに単一のページが必要だね

利用者の評価から今回実装できた機能の中に利用者から良かったという意見がいくつかあった。検索に関しては授業が複数登録されているなか、条件が合致しているものを候補出しすることが良かったようだ。授業の曜日や時間の表示だけでなく、授業のある日付を表示させることも利用者が求めていることがわかった。日付の表示と一緒に閲覧できるようにカレンダーを表示することも利用者は求めている。レコード削除については削除について確認を取るようにワンクッションおいてほしいという意見が多くあった。これら利用者から得ることのできた評価を踏まえて本研究の中で足りなかったものを明確にしていきたい。

5-4. 考察

本研究の目的である学生の授業への出席状況をデータ化し自己管理によって出席率の向上、意識の改善を促すという目的を達成することはできなかった。システムは授業の登録、登録した授業の全件を表示、条件を指定して登録した授業情報の検索、閲覧まで実装する事ができた。今回実装する事ができなかったプログラムはアカウント管理、ログインフォーム、出欠の登録、集計、授業ごとの出力である。これらが実装できた場合、学生が自分のアカウントを登録し、自分の履修授業を登録し、出欠を授業ごとに入力することによって、自分の出欠状況を正確に記録することができる。これによって学生が出欠状況を把握できるようになり出席を促すことへ繋げることができる。

利用者の評価を踏まえて考察を行っていく。登録した授業に曜日と時間以外に日付も表示させることによって授業のある日がわかると良いということ、カレンダーと一緒に表示させることで授業以外にもスケジュール管理として利用できるのではないかとということも考えられた。レコード削除について、クリックして即削除されるのではなく、確認を求められるようにワンクッションおくと良いようだ、利用者側からしてみると削除を行う際には確認画面がほしいという人は多かった。全件表示に関して、登録した授業が多くなってくると見づらくなる。そのため見づらさを解消するために授業の曜日、時間ごとに表のように閲覧できるようにすると良いのではないかと考える。

6. まとめ・今後の課題

本章では今までの議論を踏まえ実装した機能、評価をまとめ、今後の課題を述べていく。

6-1. まとめ

本研究では学生の授業出席の向上、意識改善を目的とした出欠情報をデータ化し自己管理できる機構の構築を行うものである。学生へアンケート調査を、先生へ取材調査を行い、問題点を明らかにした。その問題を解決することにより学生の出欠状況の確認をできるシステム作りを行った。授業の登録、検索、登録情報を全て表示することができたが、肝心の出欠を登録、集計、表示することはできなかった。利用者から様々な評価を得る事ができた。これを元に実装できなかった部分を実装し、利用者からの要望を反映することにより出欠自己管理機構として学生の出席意欲が高まり、学生生活の意識改善がされることにつながる。

6-2. 今後の課題

今回、授業情報の登録、登録した授業情報を全て表示、登録した授業情報を検索する機能を実装することができた。自己管理システムの機構なのに利用者一人一人を対象とするためのログインフォームの作成、アカウント管理、登録した授業ごとの出欠の入力、集計、表示をする機能を実装する事ができなかった。これらの未実装部分を実装し、学生利用者の出席向上へのさらなる支援ができるものとして完成へ近づけるようにすることが必要である。

参考資料

【1】 たにぐちまこと『よくわかる PHP の教科書』株式会社毎日コミュニケーションズ
2010年

【2】 ふろぐらむる一む

<http://plog.pya.jp/index.html>

2011年12月29日

【3】 PHP の学習なら PHPBook

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

2011年12月29日

【4】 基礎から解る！PHP 入門講座

<http://w1.nirai.ne.jp/freeze/>

2011年12月29日

【5】 PHP Web リファレンス

<http://bg.pi-ppi.com/>

2011年12月29日

【6】 西沢 直木 『PHP による Web アプリケーションスーパーサンプル』第2版

ソフトバンククリエイティブ；第2版 2006年3月24日

【7】 Rasmus Lerdorf(著) Kevin Tatroe (著) Peter MacIntyre (著) 高木 正弘 (翻訳) 『プログラミング PHP 第2版』オライリー・ジャパン；第2版 (2007年10月23日)

【8】 小島 まさご (著) 『実践マスター PHP+MySQL PHP5対応』ソーテック社 2011年11月19日

謝辞

本研究において、常に丁寧に指導して下さった指導教員の渡辺恭人准教授には大変感謝しております。渡辺恭人准教授には論文の執筆、プログラミングに至るまで丁寧に指導していただいたおかげで、卒業論文を完成させることができたと感じております。渡辺恭人准教授には、研究とは目的や必要性、問題の発見と解決方法の検討が大事であるということ、問題発見のための調査は非常に大切なこと、研究結果を評価することが大事であることをテーマ研究会において幾度となくご指導いただきました。本研究を通し、渡辺恭人准教授の仰っていた意味をようやく理解できたように感じます。また、取材、学生へのアンケート、運用実験に付き合っていたいただいた多くの方々、私の未熟な考えや説明を聞いてくださり、ご協力下さいました方々にお礼を申し上げます。皆様方のご協力により私の卒業論文は完成することができました。最後に、本研究に関わって下さったすべての方に改めて感謝を述べさせて頂き、謝辞とさせていただきます。