

政策情報学部における
コラボレーションの促進に関する検討

担当教員名 渡辺 恭人

学籍番号 0740078

氏名 連記 吉邦

概要

1. 背景と目的
2. 現状と問題点
 - 2-1 大学生の一日の勉強時間の低下
 - 2-2 現状の学校生活におけるコラボレーションの機会
 - 2-3 企画への参加者を募集する方法
3. 解決法の検討
 - 3-1 情報の共有
 - 3-2 効率的な募集、参加方法
4. 設計
 - 4-1 求められる要件
 - 4-2 システム構成
5. 実装
 - 5-1 使用ハード、ソフト
 - 5-2 画面
6. 評価
 - 6-1 動作確認
 - 6-2 類似サービスとの比較
 - 6-3 考察
7. 参考文献

1. 背景と目的

1-1 背景

政策情報学部は大学維新への挑戦千葉商科大学政策情報学部 10 年目の報告によると「現代社会が当面する問題を発見し、問題を解決するための制作を立案し、実行し、解決し、評価することをねらいとして、これまでさまざまな個別科学や学問から、また、現実社会から必要な理論や知識を学び、それらを統合化して新しい理論や知識を確立しようとする、極めて実践的で進化し続ける学問が、政策情報学である」「社会のさまざまな分野の問題発見・問題解決に取り組むことのできる人材の育成を図る」と書かれている。つまり新たな問題を発見し、解決をするためさまざまな分野とコラボレーションすることにより解決することのできる人材の育成を目的に設立された。

現状では企画制作実習やインターネット放送実習などで行われているがコラボレーションが十分ではなく活発化すべきである。そこで企画を思いついても人を集めるのが困難であることと、自分が参加したい企画がうまくマッチングすることが難しい。

企画者からの視点

企画を実行させるには何より人を集めるのが大事である。それもできればその企画と同じような事をした経験者が必要だろうが、現状では友人や友人からの紹介でなければ人を集めることは難しい。また専門的な企画の場合人を集める時間がかかり企画そのものが頓挫する可能性すらありえる。

参加者からの視点

自分のしたいことはある程度決まっているがそのような企画に出会うことがない、そもそも学校全体でいまだのような企画があるのか把握しマッチングさせるのはとても難しい。

1-2 目的

これを解決するため本研究ではあらかじめ知識や技術を持った人、企画者を収集し、分類化することにより参加者と企画者をマッチングさせ迅速に必要な人材を集め、参加する仕組みを検討しコラボレーションの活性化を目指す。

2.現状と問題点

2-1 大学生の一日の勉強時間の低下

埼玉大学の埼玉大学生に対しておこなわれた調査[6]によると大学生の一日の勉強時間が一時間以内という回答が約70%とかなりの割合を占めた。

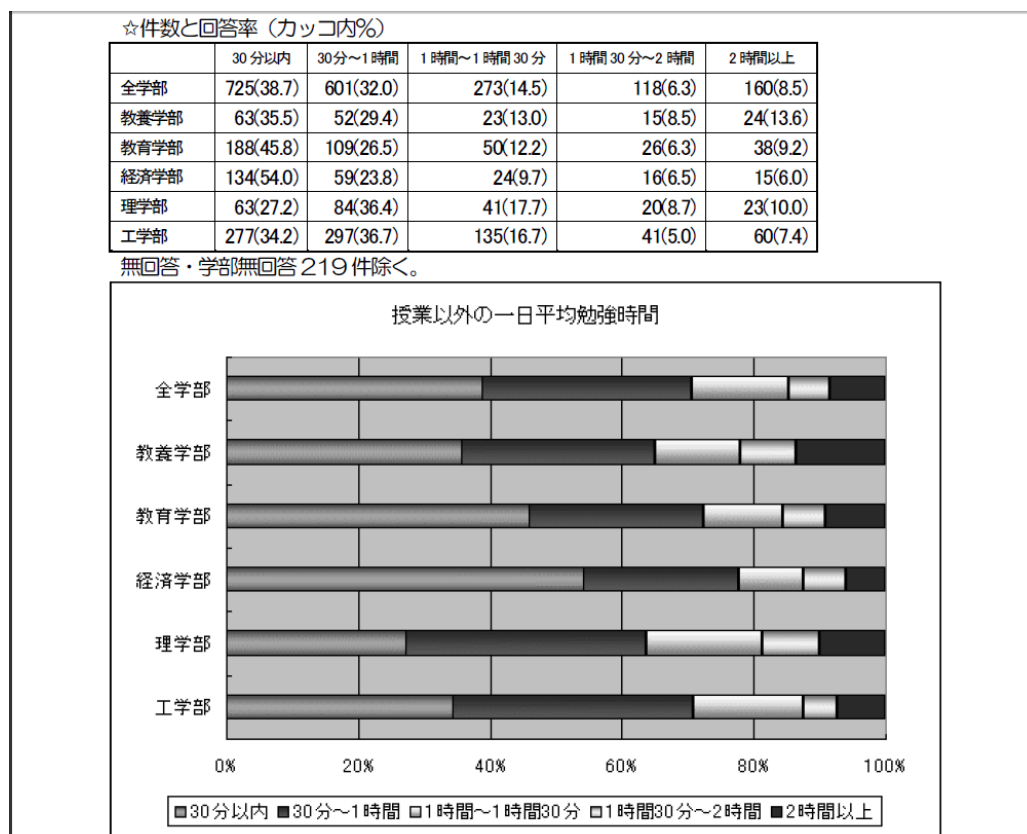


図 2.1(埼玉大学 平成20年度学生アンケート調査報告書より参照)

大学に入学するということは将来に対してのある程度のビジョンがあるはずだが、なぜこのようなことがおきたのだろうか。それは大学に来てやりたいことに対する参加する機会と情報を取得する手段が少ないからであると考ええる。

2-2 現状の学校生活におけるコラボレーションの機会

背景と目的で前述した通り政策情報学部では新たな問題を発見し、解決をするためさまざまな分野とコラボレーションすることにより解決することできる人材の育成を目的に設立された。つまり学生が主体となって企画を提案し実現、解決することが本学部の最大の目的といえる。

本学には企画制作実習といったコラボレーションを促進すると言った講義があり受講生が企画者、参加者として企画を提案、発表し実現するというものがあるが以下のような問題点がある。

問題点

1 講義内での企画者と参加者の需要の不一致

この講義では様々な種類の企画が出される。音楽、映像、ゲーム、サービス、ボランティア…またはそれ以外の何かなど多種多様である。そういったなか、受講生内でその企画にあった参加者を必要人数集めることは難しい。

企画にもよるが必然的にその企画の経験者、その企画にあった知識技術を持った参加者が必要。そのような人材をこの講義の受講生内で集めるのは難しい。なぜならこれは単位がかかっており参加者より企画者のほうが有利だからである。

1 企画者はあらかじめ企画を考えている

2 参加者が参加したい企画がない場合妥協、自ら企画を構想しなくてはならないが

企画者は参加者がいなくても募集すればいい

3 講義であること

講義であるがゆえに年二回の履修登録締め切り前に企画、あるいは自分が参加したい企画を見つけなければいけない。そのためコラボレーションの機会が限られてくる。

2-3 参加者を募集する方法の問題

企画を成功させるには前述したとおりその企画にあった人材を集めることにある。それもできれば早く集めたい。では現状ではどのような集め方があるか。

1.友人に参加を求める

2.友人またはその友人からの紹介

3.放送して参加者を求める

4.学校の掲示板に貼り出す

5.関連する団体、ゼミに募集（要アンケート）

6. メールでの送信

恐らくこの6つになるだろう。だが以下のように問題点がある。

問題点

必要人数を集められるか 全項目（おもに 1、2、5）

1 と 2 について企画者、参加者の友人に頼らざるに負えない。まず都合よくその企画にあった友人がいるかである。恐らく当てはまる友人がいるだろうがその企画の必要知識、技術を持った人材を友人だけで必要人数に集められるだろうか。

5 ゼミに募集をかけるのは1～6の項目で最も理想的な募集方法であるがその企画に賛同してくれるか、必要人数に満たない場合どのように募集するかなどの問題を抱えている。

しかし 1、2、5 は必要人数が心配なものの速く人材を集めるという点で優れている。

情報媒体の仕様方法が明確ではない 3、4

3、4 といったものも宣伝方法の一つとして考えられるがこれを利用する際の必要な手続きが一部の学生を除きわからないだろう。

またこのような手段を使う場合敷居が高くなってしまい逆に参加者が集まりにくい

可能性がある。

人材を募集する際時間がかかり過ぎること 3, 4

3、4 に関しては手続き上の問題もあるが放送内容、掲示するチラシなど作らなくてはならないため時間がかかる。

6、4 は募集しても気づかれない恐れがある。

4 大体の生徒は自分が所属する学部情報などが記載されている掲示板を主に確認するが4で掲示されるのは恐らくクラブ活動やサークル募集欄である。サークルなどのクラブ活動を求める欄ではたして気づかれるだろうか

6 メールで募集をかける場合まず受信者はそのメールを見るかどうかである。ある日知らないアドレスからメールを受信した場合スパムやウィルスを警戒して受け取られないことがある。

募集または知人から紹介してくれた人物が企画に対して興味があるか

一番の問題はこれである。自分の企画のジャンルに興味がないひとに対して勧誘しても参加する可能性は極めて低いし効率的ではない。

参加者希望者が現在行われている企画を把握できない

企画に対して参加意欲がある参加希望者は上記の方法で企画を把握しなければならないがこれは受動的であるため積極的に活動することができない。現在このような企画があって参加者を募集しているかどうかまとめ、参加希望者が自分で探しに行ける方法が必要である。

問題点のまとめ

1 企画者と参加者のニーズがあわない

現状の問題点として前述した通り現在企画の提案の場所が限られており企画者は企画参加者の募集しにくく、参加者は自分がやりたい企画を探しにくい状態である。

2 企画の宣伝方法

企画の宣伝方法も既存のものだと時間がかかり過ぎるうえ何よりその企画にあった技術、経験者、または興味を持った人材を集められるかどうか未知数である。

3 参加者の募集方法

参加者の宣伝方法が限られており、現状では企画者が能動的に参加者を募集するのに対して、参加者は能動的でありまた現在この学校でどのような企画があるのか確認する方法がないため積極的に参加しにくい。

3.解決方法の検討

これらの問題を解決し本研究で目指すシステムの全体像を検討する。

解決方法

3-1 情報の共有

現状と問題点の参加者を募集する方法の問題点 1、2、3 は共通の問題点がある。それは情報網が貧弱であることだ。

2 は自分がやろうとしている企画のジャンルに対して興味ある人が分からず効率的な勧誘ができず、3 はいまどのような企画があつてどのような人を募集しているのかが分からない。それが1のようなニーズのミスマッチを生み出している。

つまり企画者は自分の企画のジャンルに興味がある人を把握でき、参加者も現在行われている企画を把握ができやりたいジャンルの企画に参加、ニーズのマッチングすることによってこの問題は解決できる。そのためには双方の情報が伝わる仕組みが必要である。

類似サービス

企画者、参加者が把握し合うことでこの問題は解決することが分かった。ではどのような方法で参加、募集するべきか類似サービスを調査し分析することによって本研究の目指すものを検討する。

評価方法として以下の点を評価する。

1 サイト名 JREC-IN <http://jrecin.jst.go.jp/seek/SeekTop>



図 3-1JREC-IN

概要：JREC-IN は独立行政法人科学技術復興機構が運営するサイトで、研究機関が研究員を公募するサイト。本サイトの概要によると「JREC-IN は、研究に関する職を希望

する求職者情報と、産学官の研究に関する求人公募情報をそれぞれ収集・データベース化して、インターネットを通じて無料で提供しています。」と書かれている。

表 3-1 JREC-IN の評価

	企画提案	参加	検索	企ジャンル分け	登ジャンル分け
評価	不明	○	◎	○	○

1 企画が提案しやすいか

企画を提案するには会員登録が必要なのだが、利用者条件に当てはまらなかったためどのように登録しているか不明である。

2 参加しやすいか

募集要項などが分類化され詳細に書かれているため参加しやすくなっている。

また募集要項とプロフィールが合致した場合マッチングメールが送られる仕組みに立っている。

3 検索機能があるか

検索機能が実装されておりキーワード、期間種別、研究分野、勤務地、職種で検索できるので非常に優れている。

4 企画がジャンル分けされているか

されている。

5 登録者がジャンル分けされているか

会員登録時に自分の専攻や研究によってジャンル分けされる。HPによると登録者のプロフィールなどが見れるらしいが利用者条件に当てはまらなかったのではわからず。

メリット

このサイトの良い点は検索機能のオプションが多いことだ。会員制にすることで求職者のスキルを登録し求人者と求職者をマッチングした場合お知らせメールを届けることでマッチングを促している。

デメリット

求人ページに現在募集しているかどうかのステータスがないためいちいち問い合わせる必要がある。

2 サイト名 バンドメンバー募集 OURSOUNDS<http://mb.oursounds.net/index.html>



図 3-2 OURSOUNDS

概要：バンドメンバー募集サイト。会員登録をする際に、都道府県（市長区村まで）を入力することで全国に対応している。また募集だけではなく参加希望を出してバンドに参加することができる。

表 3-2 OURSOUNDS の評価

	企画提案	参加	検索	企ジャンル分け	登ジャンル分け
評価	◎	◎	◎	○	○

1 企画が提案しやすいか

非常に提案をしやすい。活動地域、活動曜日、性別、年齢、募集パートなど入力する情報が多いだけではなく、自分の記事をHP上で管理編集ができるので管理しやすい。

2 参加しやすいか

バンド募集要項が詳細で確認しやすく、参加希望を知らせるのにメールではなくHP上でメッセージを送れるので非常に参加しやすい。またバンド参加希望で記事を投稿できるため能動的に動くことが可能。

3 検索機能があるか

実装されている。担当パート、好みのジャンルや活動地域などでメンバー募集、参加希望両者を検索できる。

4 企画がジャンル分けされているか

音楽のジャンルと募集メンバーのパートでジャンル分けされている

5 登録者がジャンル分けされているか

されている。登録時に性別、年齢、このみの音楽ジャンルでジャンル分けされている

メリット

バンド募集ページにどのようなメンバーを募集しているか、ジャンルは何か書かれており活動地域、曜日も書かれていてとても見やすい。

また投稿した記事を編集するページがあり管理がしやすい。参加希望者が出た場合メッセージボックスがありそこで確認できるためいちいちメールサーバーを見なくてもよいので手間が省けている。

バンドメンバー募集だけではなく、自分はこのようなスキルがあってこのようなジャンルに興味があるという記事を投稿してメンバーに入れてもらう仕組みが整っている。そのため企画者、参加者のニーズがマッチングしやすい。

デメリット

このようにバンドのジャンルとバンド参加希望者のジャンルがあるのにマッチングメールのような仕組みがないため、本来マッチングできるのに見逃している場合がある。

3 サイト名 スポーツやろうよ! <http://www.net-member.com/>



図 3-3 スポーツやろうよ！

概要：スポーツサークル専用のメンバー募集サイト。球技、格闘技、ウィンタースポーツ、マリンスポーツなど多く種目がある。

表 3-3 スポーツやろうよ！

	企画提案	参加	検索	企ジャンル分け	登ジャンル分け
評価	○	○	●	○	●

1 企画が提案しやすいか

企画を提案できるが管理ページなどが無いので管理しにくい。

2 参加しやすいか

参加は直接メールでの連絡するため参加しやすいというわけではない。また活動地域

で検索が不可能なので検索がしにくい

3 検索機能があるか

実装されていない。

4 企画がジャンル分けされているか

ジャンル分けがされているが地域で分けられていないので見難い。

5 登録者がジャンル分けされているか

ジャンル分けされていない。

メリット

非常にシンプルに構成されているため記事を投稿しやすい。スポーツの種類によってジャンル分けされているため目的にあったスポーツの募集記事を見ることができる。

デメリット

会員登録制ではないため企画者が参加希望者を検索する方法がない。また地域や活動曜日などで検索ができないため無駄が多い。

4 サイト名 ふりーむ！ <http://www.freem.ne.jp/cgi-bin/bbs/party/osiete.cgi>

図 3-4 ふりーむ！

概要：フリーゲームを集めたサイトのコンテンツのひとつゲーム開発募集掲示板。

表 3-4 ふりーむ！

	企画提案	参加	検索	企ジャンル分け	登ジャンル分け
評価	○	●	●	●	●

1 企画が提案しやすいか

提案しやすいが募集メンバーを直接入力しなければならない。

2 参加しやすいか

参加方法は投稿者が決めているので参加しにくい場合がある

3 検索機能があるか

検索機能はない

4 企画がジャンル分けされているか

ジャンル分けされていない

5 登録者がジャンル分けされているか

ジャンル分けされていない

メリット

特になし

デメリット

企画も募集メンバーもジャンル分けしていないので参加希望者タイトルを見て判断するしかない。また連絡を取る場合の方法も企画者によって違うので手間が多い。

まとめ

表 3-5 評価まとめ

サイト名	企画提案	参加	検索	企ジャンル分け	登ジャンル分け
JREC-IN	不明	○	◎	○	○
OURSOUNDS	◎	◎	◎	○	○
スポーツ やろうよ	○	○	●	○	●
フリーム	○	●	●	●	●

1 企画提案

◎入力項目が 7 項目以上○入力項目が 6~4△入力項目が 3~1●実装されていない

2 参加

◎参加者の募集項目があり参加希望者を管理する仕組みがある○募集項目があり連絡先を入れる項目がある△募集項目がある●直接入力

3 検索機能

◎検索項目 5 項目以上○4~2 項目△1 項目●実装されていない

4 企画がジャンル分けされているか

○されている●されていない

5 登録者がジャンル分けされているか

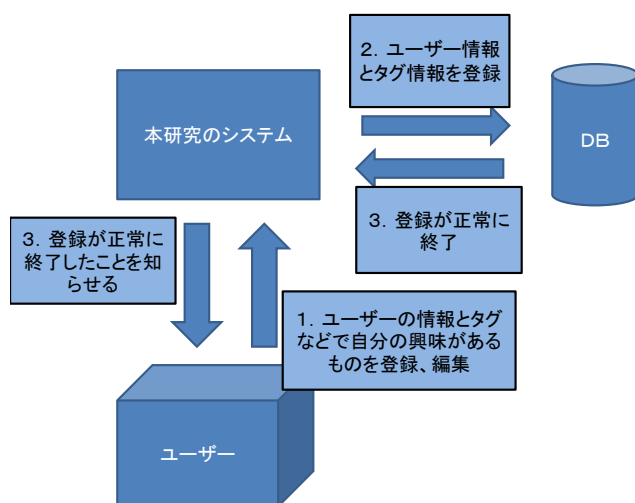
○されている●されていない

調査した結果 OURSOUNDS と JREC-IN が高評価を得た。2 つに共通しているのは会員サイトであることと企画登録、会員登録時に詳細情報を入力しお互いに情報を共有することで何を求めているのか、何ができるのかをわかりやすくしたことにある。また JREC-IN のマッチングメールシステムは参加者と企画者のマッチングを促進するのに最適だと言える。

3-2 効率的な募集、参加方法

情報を共有することでお互い能動的に活動することができ効率的であることが既存のサービスを分析した結果わかった。

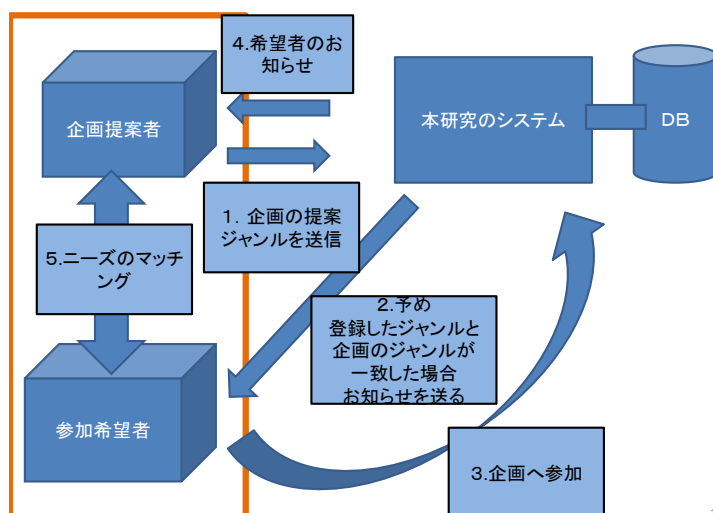
- 1 企画投稿、会員登録時に詳細情報を入力し共有する仕組み
 - 2 そのためにも会員サイトであること
 - 3 登録された企画の募集要項と参加者の情報がマッチングしたら知らせるシステム
- 効率的な企画のマッチングさせる方法はこの三つをクリアしなければならない。
そこで図1、図2のような情報モデルが望ましい



15

図 3-5 会員登録時

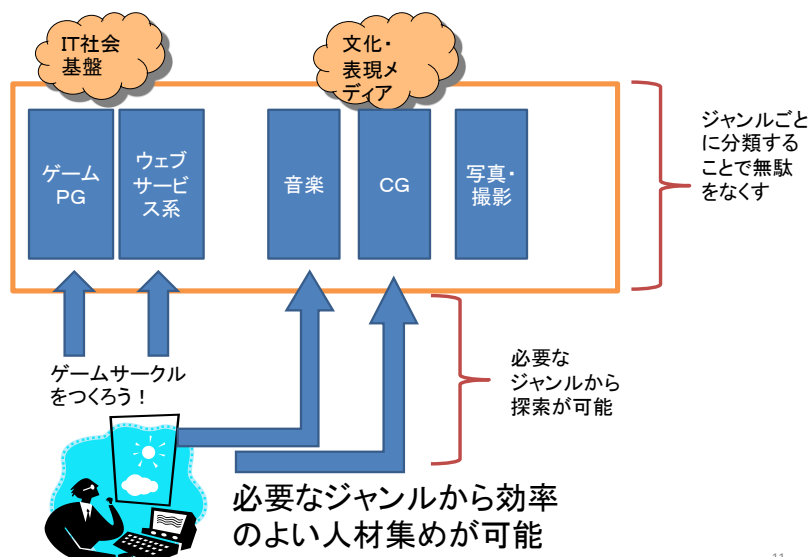
図1はユーザーの情報タグなどをデータベースに入力するまでの流れだ。ユーザー情報を入力する。



14

図 3-6 企画の提案から参加までの流れ

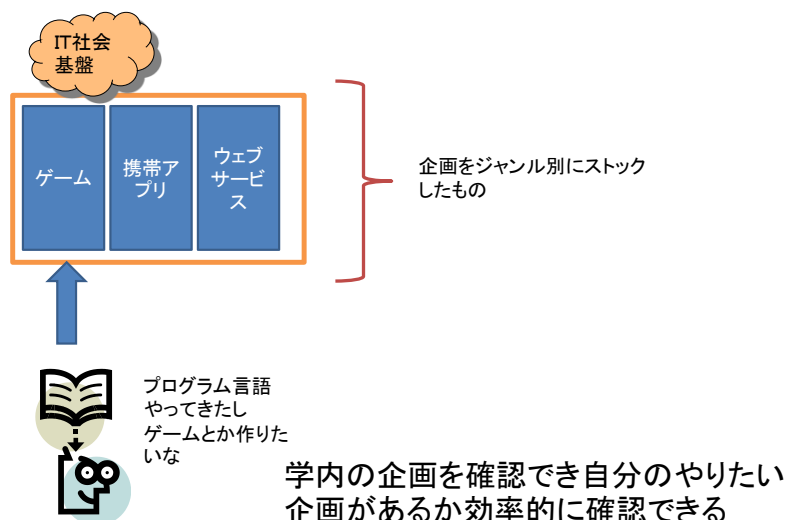
図2は企画を提案しマッチングされるまでの流れだ。1で企画のジャンルなどを登録して企画のジャンルとユーザーのジャンルがあった場合紹介する。
この情報モデルにより既存のサービス分析結果1、2、3をクリアすることができる。
例えばこのような情報モデルでマッチングシステムを作った場合図3のようなことができる。



11

図3-7 ゲームサークルを立ち上げる場合

この図のようにユーザーをジャンルで収集することで効率的に人を集めることができる。また興味を持った人に対してのみ募集がかけられ効率的にメンバーを集められる。



12

図3-8 企画を検索する場合

また参加者も図2のように収集した企画から自分に合った企画を能動的に調べること

ができればマッチングを促進させることが可能である。

初期ユーザーに対しての補助について

実際に企画を運営する上で問題が発生するのは常にあることだ。特に初めて企画を運営する場合特に多い。企画運営、技術、広告など様々な問題が出てくるだろう。

それに対応するためユーザー同士が助け合う仕組みを検討した。Help 掲示板を作りユーザー同士が助け合うことで解決することにした。基本的な情報モデルは図 1 と同じで問題のジャンル分けしてユーザーに開示、それに対する答えをもらう方法だ。

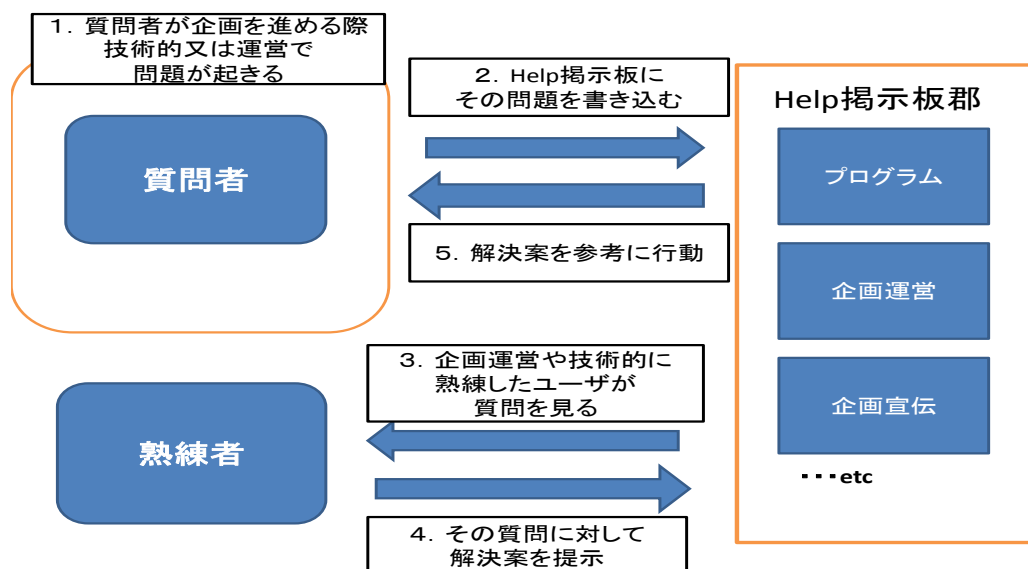


図 3-9 Help 掲示板の情報モデル

この方法のいいところはユーザー同士同じ境遇に立っているので同じ問題を過去に体験したユーザーがいる場合がある。そのためどのようにしたか体験段や技術的なアドバイスが聞けるだろう。

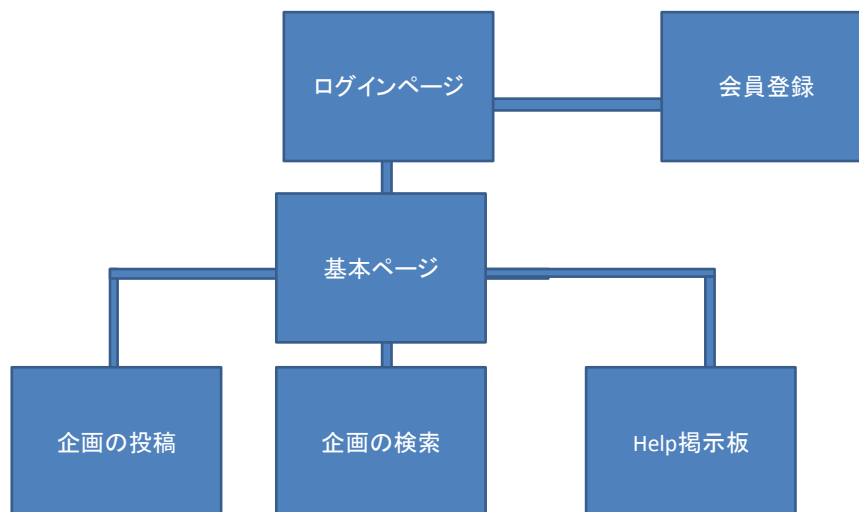
4.設計

3章に基づき設計を行う

4-1 求められる要件

1. 会員サイトで入会時に自分のスキルと興味があるジャンルを登録できること。
2. 企画を投稿できる。その場合企画のジャンルも登録できること
3. 企画のジャンルと会員のジャンルがマッチした場合会員にお知らせが届くようにする。
4. Help 掲示板がある

この4項目が本研究で求められる要件である。図3がサイト構成図である。



15

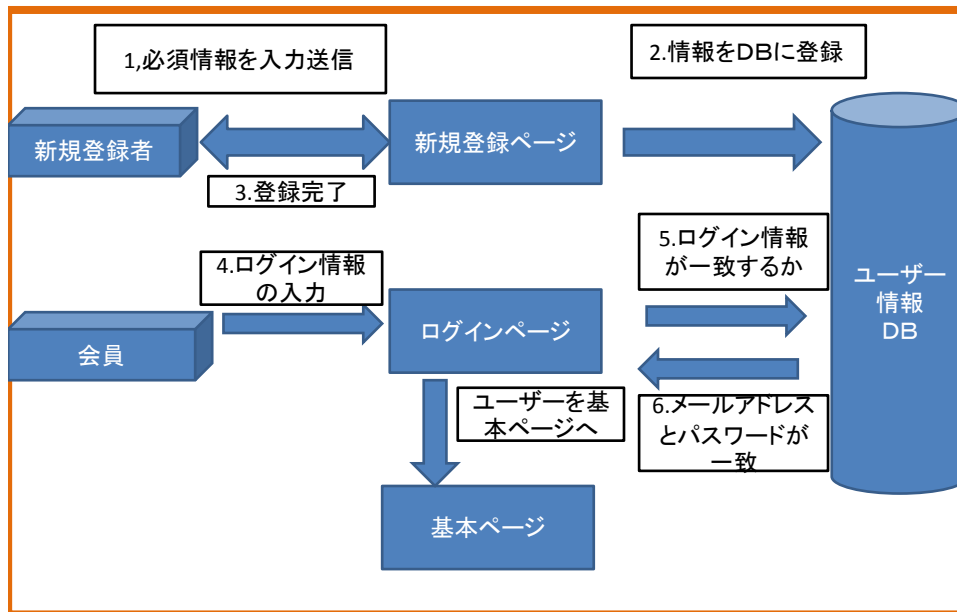
図 4-1 サイト構成図

ログインページでログインし基本ページから企画の投稿、検索、Help 掲示板に行け、会員でない場合は会員登録ページで会員登録をする。この構成図と情報モデルに基づき設計を行う。

4-2 システム構成

ログイン、会員登録

要件の一つユーザー登録の設計をする。



黄色い枠はセッションを使用

15

図 4-2 新規登録、ログイン認証

1 でユーザーがメールアドレスとパスワード、名前、興味のあるジャンルを入力し 2 でデータベースに登録する。その後 4 でメールアドレスとパスワードを入力し 5 でログインページがデータベースに対して一致するかどうか命令する。一致した場合基本ページへ飛びログインを完了する。

新規登録からログインまで一連の操作をセッションを使い入力情報を保護しセキュリティ性能を高める。

テーブルの設計

ユーザー情報を登録するデータベースは下記のように設計をした。

表 4-1 userDB

フィールド名	データ形	内容
user_name	char	ユーザーの名前
user_mail	Text	ユーザーのメールアドレス
user_password	char	パスワード
user_type	char	興味のあるジャンルの選択
user_no	int	会員 no (重複対策)

user_name と user_password で本人確認をする。パスワードを忘れた場合 user_mail をと user_name を入力してもらいから一致した場合メールを出せるように設計をした。

user_no は登録をするたびに + 1 ずつ入力される。内容を重複して登録し内容を消す場合両方のデータが消えないための対策である。

企画の投稿、検索

企画の投稿と検索システム

企画入力ページから企画のタイトル、企画内容、企画のジャンルを入力し KikakuDB に入力する。

テーブルの設計

表 4-2 KikakuDB

フィールド名	データ形	内容
name	char	企画タイトル
naiyou	Text	企画内容
tag	char	企画ジャンル
delete_key	char	記事を消すためのキー
no	int	記事ナンバー（重複対策）

マッチングした場合の設計

企画と会員情報とのマッチングの仕方

マッチングさせるのは表 4.1 と表 4.2 の `user_type` と `tag` が一致するかを調べる。もしユーザーと企画のジャンルが一致した場合メールユーザに対してメールを送信する。

Help 掲示板

ページでジャンルを分けるので `tag` などにはつけない。スレッド方式の掲示板を目指す。スレッド方式を選択したのは企画を運営する上で発生する問題は様々なため通常の掲示板では対応できない。

記事を投稿するたびにデータベースにテーブルを作成し新たな掲示板を作ることでスレッド方式の掲示板を目指す。

テーブルの設計

表 4-3HelpDB

フィールド名	データ形	内容
name	char	掲示板のタイトル
response	text	書き込み内容
Delete_key	char	掲示板を消すためのキー
no	int	掲示板のナンバー（重複対策）

5.実装

検証部分について

本研究では企画の登校検索システムの検証を行う。この検証部分はシステムの中核であり最も実装されなければならない部分だからである。

5-1 実装環境

クライアント側

OS

CPU

メモリ

使用ブラウザ：IE8

サーバー側

OS

CPU

メモリ

Web サーバー：

データベース：

プログラミング言語：PHP（バージョン）、HTML

表 5-1KikakuTouroku.php

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

<html xmlns="http://www.w3.org/1999/xhtml"lang="ja"xml:lang="ja">
<head>
    <title>新規登録</title>
<meta http-equiv="Content-Type"content="text/html;charset=UTF-8"/>
</head>
<body bgcolor="#00CFFF">
<form action='KikakuTourokuKanryou.php'method="POST">
あなたの企画をD Bに登録しましょう。<br>
登録することでH Pでメンバーの募集ができます<br>
**必ず最後に連絡先と募集メンバーを書きましょう**
<br>
企画名
<br>
<input type="text"name="name"size="50"value="">
<br>
```

内容

<textarea name="naiyou" rows="20" cols="50"></textarea>

ジャンル登録をしてください（検索時に使います）

<input type="checkbox" name="tag[]" value="1">PG

<input type="checkbox" name="tag[]" value="2">CG デザイン

<input type="checkbox" name="tag[]" value="3">音楽

<input type="checkbox" name="tag[]" value="4">シナリオライター

<input type="checkbox" name="tag[]" value="5">HP 制作

<input type="submit" value="登録します">

</form>

</body>

</html>

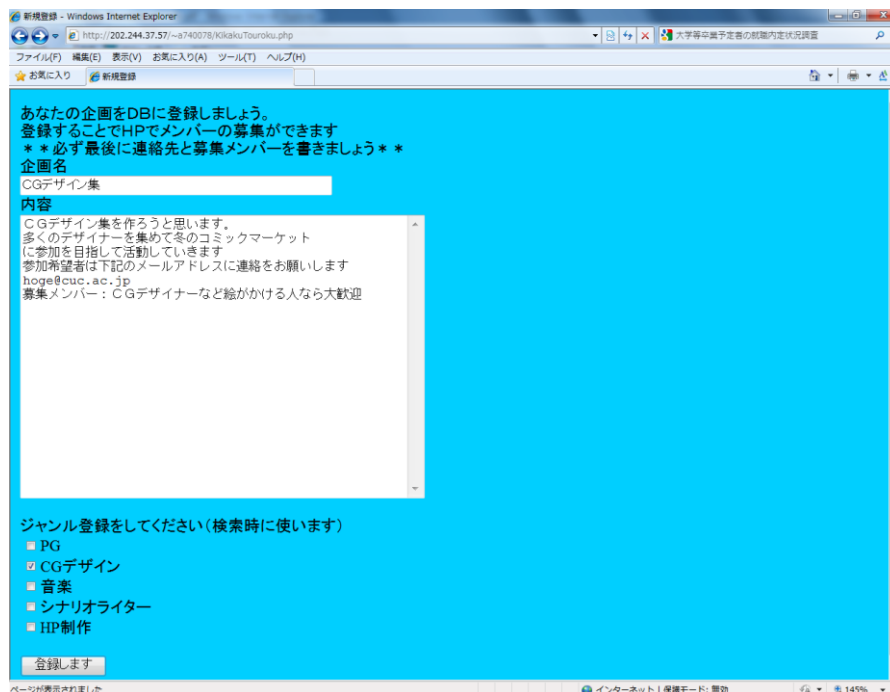


図 5-1 KikakuTouroku.php

KikakuTouroku.php で KikakuTourokuKanryou.php にデータを送信する

表 5-2KikakuTourokuKanryou.php

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

<html xmlns="http://www.w3.org/1999/xhtml"lang="ja"xml:lang="ja">
<head>
    <title>ラジオボタン</title>
<meta http-equiv="Content-Type"content="text/html;charset=UTF-8"/>
</head>
<body bgcolor="#00CFFF">
<?php
extract($_POST);

//echo "<br>name;";
//echo ($name);
//echo "<br>tag;";

foreach ($tag as $Ttag){

    $oldtag=$oldtag.$Ttag;
}
echo
$delete = mt_rand(10000,99999);
$naiyou = nl2br($naiyou);

mysql_connect('localhost','root','20cuc08!');
mysql_select_db('lesson_a740078');
$sql="insert into KikakuDB values('$name','$naiyou','$oldtag','$delete','0')";
$result=mysql_query($sql);
```

```

$sql="select*from KikakuDB where name='$name' AND delete_key='$delete'";
$result=mysql_query($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>";
        print "登録完了。あなたの登録 No は";
        echo $row["no"]."<br>";
        echo "delete キーは".$row["delete_key"];
        print "<br>";
        echo"企画名 : ";
        echo $row["name"];
        print "<br>";
        print"企画内容 : ";
        echo $row["naiyou"];
        echo "</p><hr />";
    }
}

//print "delete キーと登録 No を入力することで可能です<br>";
//print "忘れた場合は管理人にメールを送ってください";
?>

</body>
</html>

```

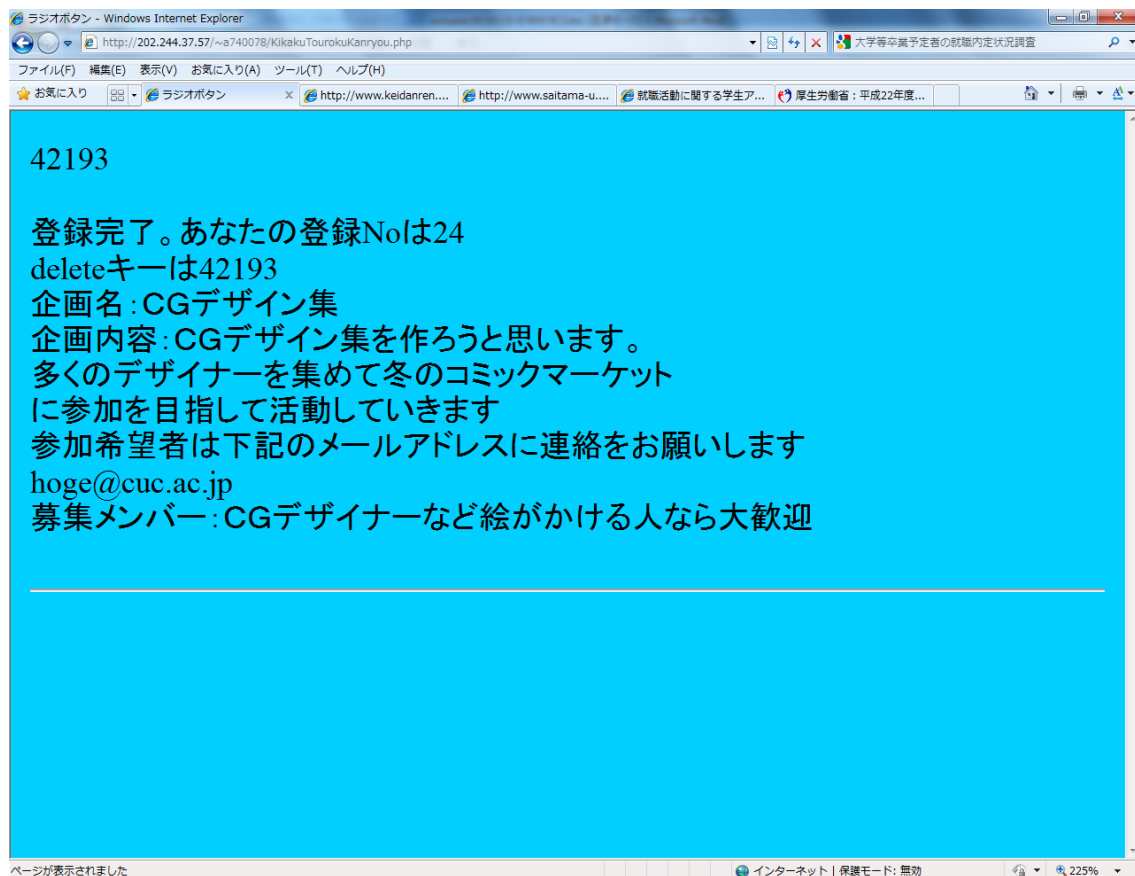


図 5-2 KikakuTourokuKanryou.php

KikakuTourokuKanryou.php で KikakuDB に情報を入力が完了。

表 5-3KikakuKensaku.php

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

<html xmlns="http://www.w3.org/1999/xhtml"lang="ja"xml:lang="ja">
<head>
    <title>全件検索</title>
<meta http-equiv="Content-Type"content="text/html;charset=UTF-8"/>
</head>
<body bgcolor="#00CFFF">

<?php

$arr[] ="";
```

```

$arr[] = "PG:";
$arr[] = "CG デザイン:";
$arr[] = "音楽:";
$arr[] = "SE:";
$arr[] = "DE:";

for($i=0;$i<7;$i++){

    $te=substr($str, $i,1);
    for($j=0;$j<7;$j++){

        if($j==$te){
            echo $arr[$j];
        }

    }

}

mysql_connect('localhost','root','20cuc08!');
mysql_select_db('lesson_a740078');
$sql="select*from KikakuDB";
$result=mysql_query($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>";
        print "登録 No";
        echo $row["no"];
    }
}

```



```

        for($i=0;$i<7;$i++){

$te=substr($row["tag"], $i,1);
for($j=0;$j<7;$j++){

if($j==$te){
echo $arr[$j];
}

}

}

        print "<br>";


        echo"企画名 : ";
        echo $row["name"];
        print "<br>";
        print"企画内容 : ";
        echo $row["naiyou"];
        echo "</p><hr />";

    }
}

?>

```

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

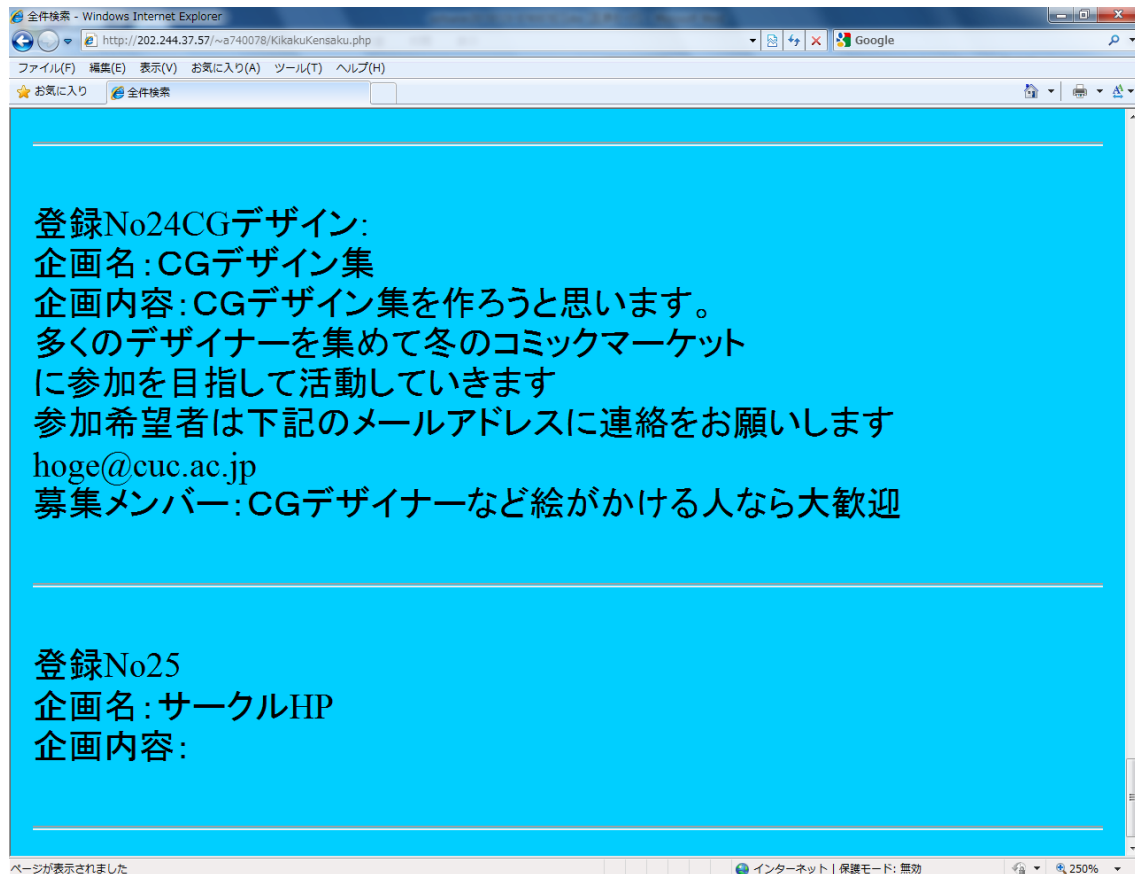


図 5-3 KikakuKensaku.php

このページで KikakuDB に入力されたデータをすべて表示する。

表 5-4KikakuTagKensaku.php

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

<html xmlns="http://www.w3.org/1999/xhtml"lang="ja"xml:lang="ja">
<head>
    <title>検索</title>
<meta http-equiv="Content-Type"content="text/html;charset=UTF-8"/>
</head>
<form action='KikakuTagKensakuKekka.php'method="POST">
企画名
    <br>
    <input type="text"name="name"size="50"value="">
    <br>
```

```



```

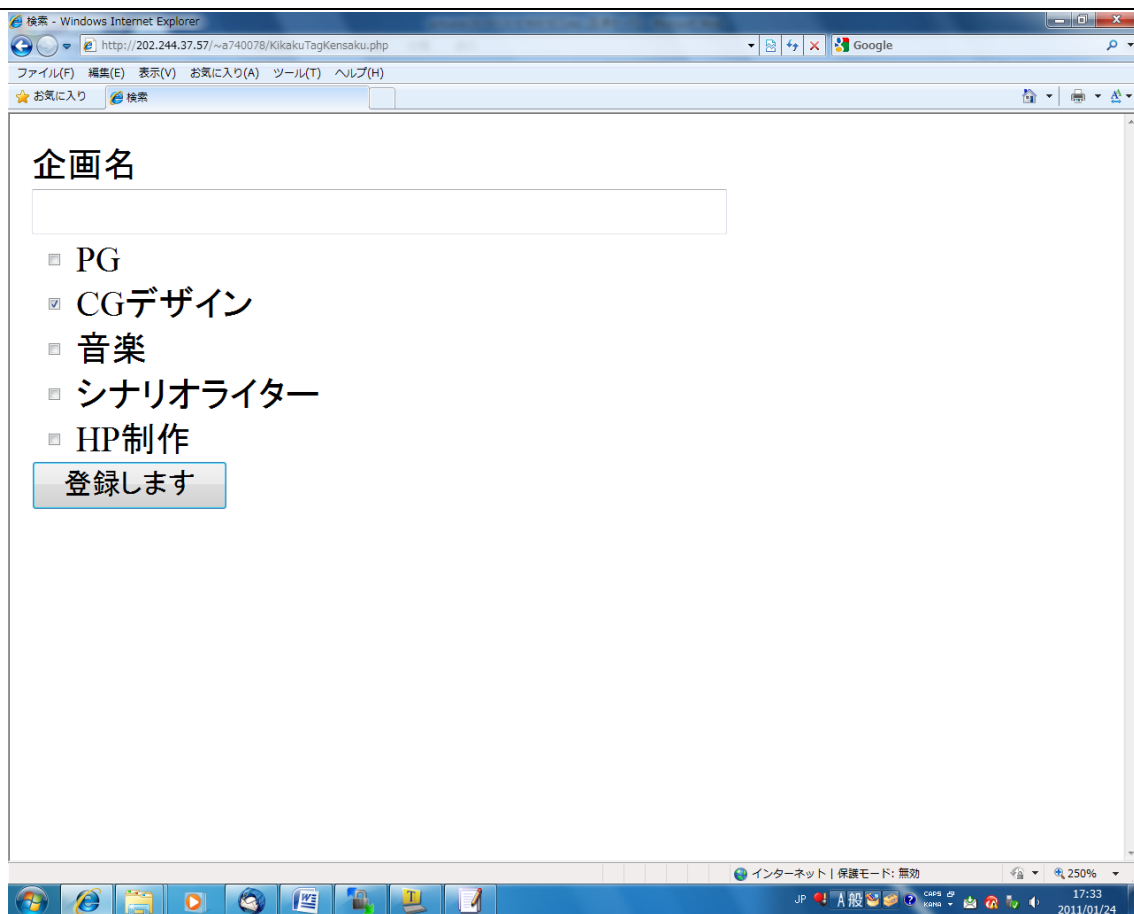


図 5-4 KikakuTagKensaku.php

検索に必要な情報を入力。ジャンルの CG デザインをチェックし検索

表 5-5 KikakuTagKensaku.Kekkaphp

```

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

<html xmlns="http://www.w3.org/1999/xhtml" lang="ja" xml:lang="ja">
<head>

```

```

<title>新規登録</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
</head>

<?php
extract($_POST);

foreach ($tag as $Ttag){

    $oldtag=$oldtag.$Ttag;
}

mysql_connect('localhost','root','20cuc08!');
mysql_select_db('lesson_a740078');


$sql="select*from KikakuDB where tag='$oldtag'";
$result=mysql_query($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>";
        print "登録 No";
        echo $row["no"];
        print $row["tag"];
        print "<br>";
        echo"企画名 : ";
        echo $row["name"];
        print "<br>";
    }
}

```

```

        print"企画内容：";
        echo $row["naiyou"];
        echo "</p><hr />";

    }
}
?>
</html>

```

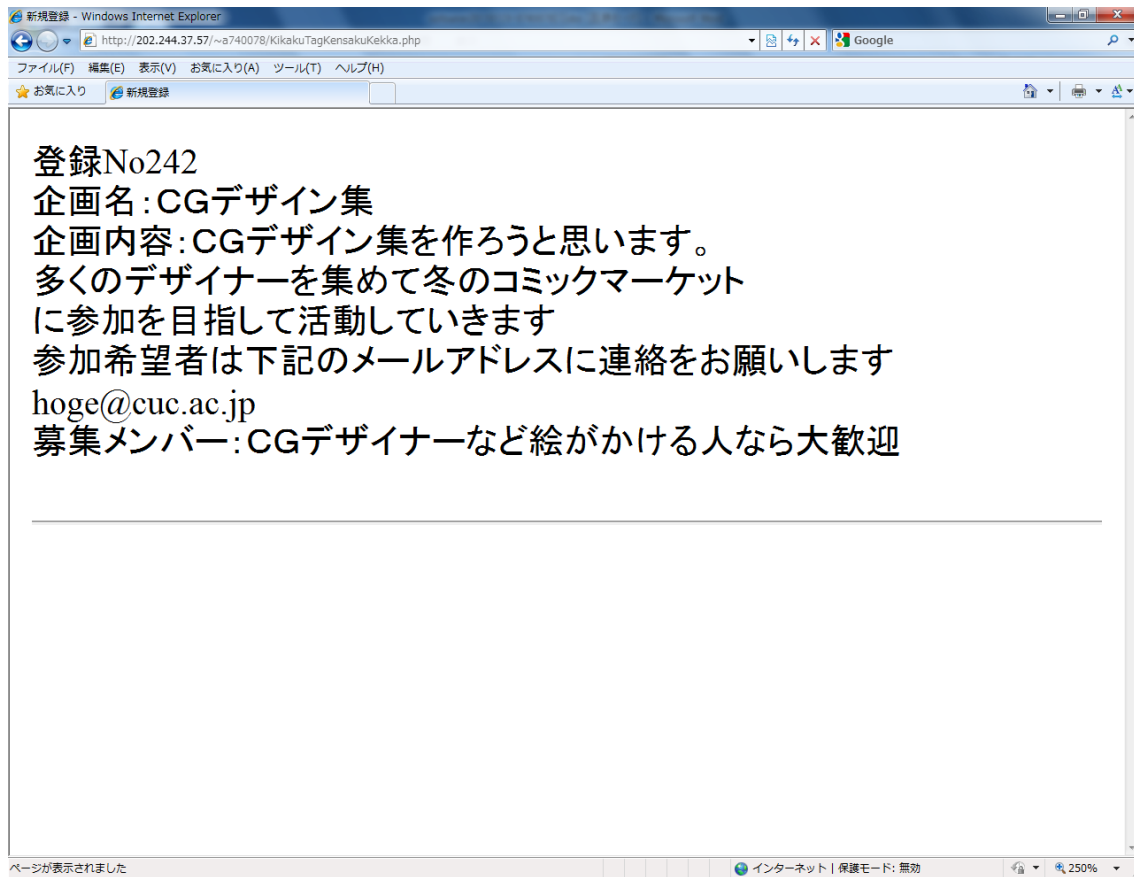


図 5-5KikakuTagKensaku.Kekkaphp

選択されたジャンルが KiakuDB にあるか検索、結果を表示する

6.評価

6-1 動作確認

本研究の検証部分の企画を登録できるか、検索できるかを確認する。検証部分は以下の通りである。

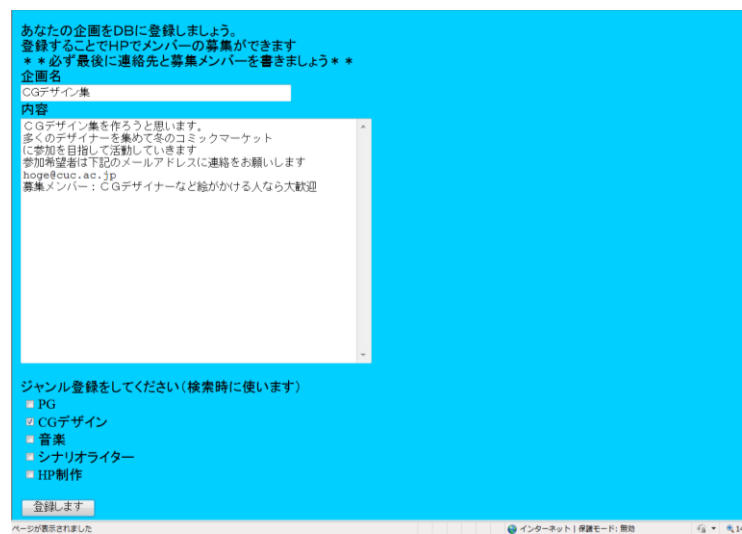
検証 1：企画を投稿できるか。

検証 2:ユーザーがその企画を検索することができるか。

まずは検証 1 から検証する。

検証 1 登録できるか

企画登録画面で必要な情報を入力



あなたの企画をDBに登録しましょう。
登録することでHPでメンバーの募集ができます
* * 必ず最後に連絡先と募集メンバーを書きましょう * *

企画名
CGデザイン集

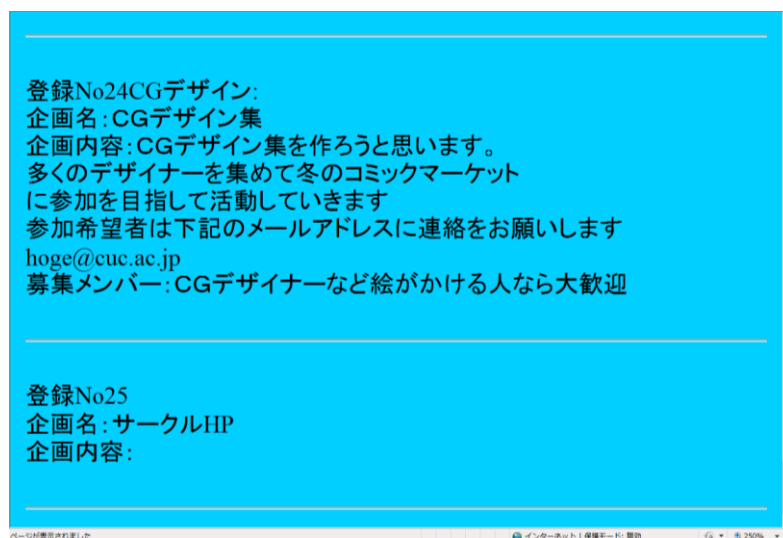
内容
CGデザイン集を作ろうと思います。
多くのデザイナーを集めて冬のコミックマーケット
に参加を目指して活動していきます
参加希望者は下記のメールアドレスに連絡をお願いします
hoge@cuc.ac.jp
募集メンバー: CGデザイナーなど絵がかける人なら大歓迎

ジャンル登録をしてください(検索時に使います)

- ☐ PG
- ☒ CGデザイン
- ☐ 音楽
- ☐ シナリオライター
- ☐ HP制作

図 6-1 企画入力

KikakuKensaku.php で登録を確認する



登録No24CGデザイン:
企画名:CGデザイン集
企画内容:CGデザイン集を作ろうと思います。
多くのデザイナーを集めて冬のコミックマーケット
に参加を目指して活動していきます
参加希望者は下記のメールアドレスに連絡をお願いします
hoge@cuc.ac.jp
募集メンバー:CGデザイナーなど絵がかける人なら大歓迎

登録No25
企画名:サークルHP
企画内容:

図 6-2 登録確認

登録が確認された。

問題点

実装したプログラムで以下の問題を発見した。**KikakuTouroku.php** の入力画面で未入力的项目があってもデータベースに入力されてしまう。これでは入力ミスして登録してしまうため未入力項目がある場合データベースに入力できなくしユーザーに足してどこの入力項目が未入力か知らせる仕組みが必要である。

検証 2 検索できるか

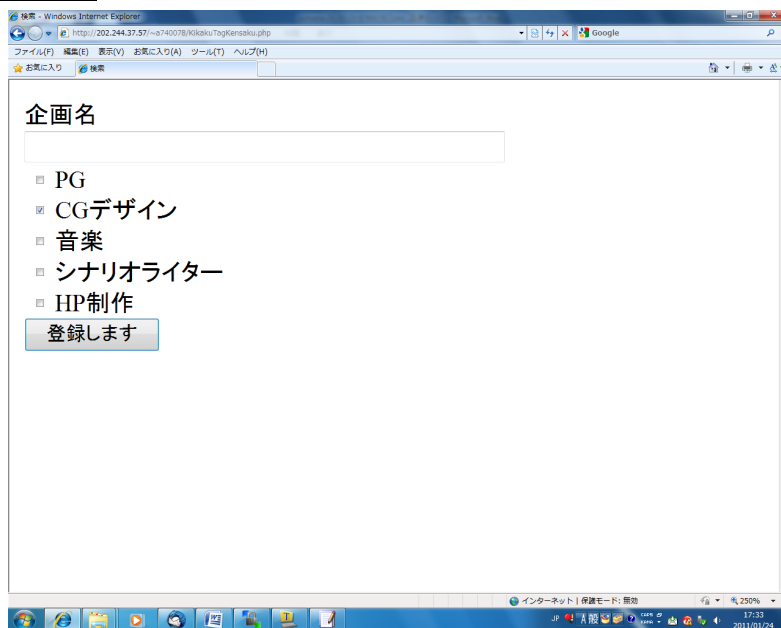


図 6-3 検索項目入力

ジャンルでの検索を行う。CGデザインで登録したためCGデザインチェックし検索

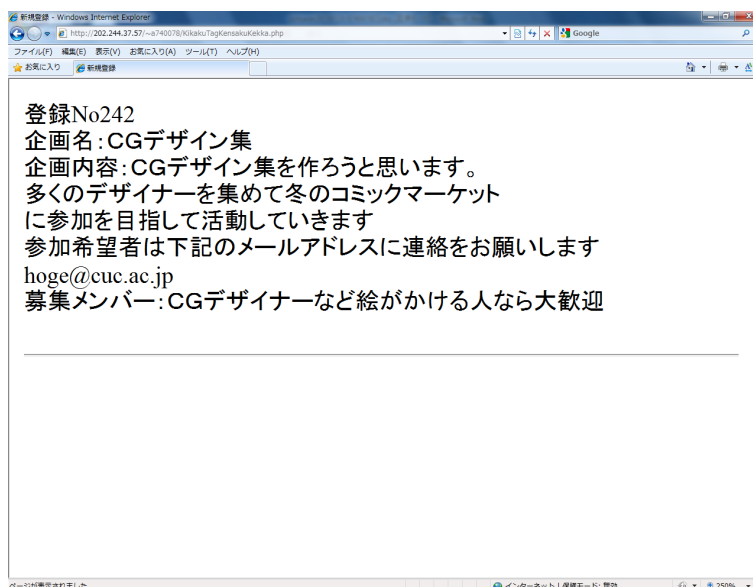


図 6-4 検索結果

ジャンル検索ができることを確認した。

問題点

KikakuTagKensaku.php での検索で企画名を入力する項目があるが実装されていない。そのため検索する情報が少なくなっている。

6-2 類似サービスとの比較

類似サービスと比較する。

本研究の検証部分では企画の登録、参加、検索、企画のジャンル分けが成功した。しかし問題点として参加申請が企画者によってまちまちになってしまうこと、検索方法がジャンルのみを考慮すると不十分といえる。そのため△とした。

まとめ

表 6-1 実装部分の比較

サイト名	企画提案	参加	検索	企ジャンル分け	登ジャンル分け
JREC-IN	不明	○	◎	○	○
OURSOUNDS	◎	◎	◎	○	○
スポーツ やろうよ	○	○	●	○	●
フリーム	○	●	●	●	●
実装部分	△	●	△	○	●

この表を見る限り既存のサービスを超えているとはまだいえない。

1 企画提案

◎入力項目が 7 項目以上○入力項目が 6~4△入力項目が 3~1●実装されていない

2 参加

◎参加者の募集項目があり参加希望者を管理する仕組みがある○募集項目があり連絡先を入れる項目がある△募集項目がある●直接入力

3 検索機能

◎検索項目 5 項目以上○4~2 項目△1 項目●実装されていない

4 企画がジャンル分けされているか

○されている●されていない

5 登録者がジャンル分けされているか

○されている●されていない

6-3 考察

一応検証部分をクリアことにはなるがまだ不十分な点が多い。入力ミスに対する対応、検索項目の充実化などをしなければ既存の類似サービスを越えることはできないだろう。

また DeleteKey を使用した企画の削除システムが実装されていないため募集を締め切

ることができないなどの問題がでた。

問題に対しての対応

未入力項目があっても入力される問題に足しては未入力項目があった場合エラー処理を行いユーザーに対してどこが未入力か知らせる。

企画名検索ができないのを検索できるようにする。

7. まとめと今後の課題

7-1 まとめ

本研究では学内の企画を収集し提示するシステムを検討し類似サービスをサーベイすることで必要機能を調べ、設計し企画登録、検索部分を検証し動作することが確認できたが新たな問題としてヒューマンエラーに対する対応と検索機能に企画名検索ができないことが出てきた。

7-2 今後の課題

1 動作確認で発見した問題を解決する。

登録画面で未入力項目がある場合ユーザーに知らせる仕組み、企画名での検索を実装する。

2 検索項目の充実化

検索項目を増やしマッチングを促す。

3 未検討の個所の部分の実装と評価

ログイン、会員登録、マッチングした場合のお知らせ、Help 掲示板を実装することで既存の類似サービスに匹敵するシステムを目指す。

参考文献

- [1] P H P 逆引きレシピ (株式会社翔泳社)
- [2] 大学維新への挑戦千葉商科大学政策情報学部 10 年目の報告 (中央公論新社)
- [3] CSS レイアウト実践講座
<http://css.uka-p.com/>
- [4] そふいの P H P 入門
<http://nyx.pu1.net/>
- [5] 日本経団連 新卒採用 (2010) に関するアンケート調査
<http://www.keidanren.or.jp/japanese/policy/2010/030kekka.pdf>
- [6] 埼玉大学 平成20年度学生アンケート調査報告書
<http://www.saitama-u.ac.jp/support/houkoku20.pdf>
- [7] 株式会社 矢野経済研究所就職活動に関する学生アンケート調査 2009
<http://www.yano.co.jp/press/press.php/496>
- [8] 情報社会学会誌 Vol.3 No.2
OSS コミュニティにおけるオープンコラボレーションの理解
<http://se.naist.jp/achieve/pdf/416.pdf>
- [9] JREC-IN
<http://jrecin.ist.go.jp/seek/SeekTop>
- [10] OURSOUNDS
<http://mb.oursounds.net/index.html>
- [11] スポーツやろうよ！
<http://www.net-member.com/>
- [12] ふりーむ！
<http://www.freem.ne.jp/cgi-bin/bbs/party/osiete.cgi>

謝辞

本研究を行うに当たり、ご指導を頂いた渡辺 恭人准教授に感謝致します。

論文の構成・執筆にいたるまで親身になってご指導頂いたおかげで、私の実力以上の卒業論文にすることができました。本当にありがとうございました。

渡辺テーマ研究会ではわかりやすく PHP や SQL、HTML をご指導いただいて多くの情報技術の可能性と教えて頂きました。提出物に対しても細かい指導と明るいゼミ運営でこの大学生活をこのゼミで過ごせてとてもうれしく思います。

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。