

卒業制作成果報告書

2021 年度 デジタルハリウッド大学

 Open Class.



目次

制作者について	3
■三嶋遥樹 Misaki Haruki	4
■矢野涼 Yano Ryo	
概要	5
■ "OpenClass." とは	6
■ 詳細	
■ ターゲット	7
■ 競合するサービスとの差別化	8
経緯	9
■ なぜ私たちが "OpenClass." をつくったのか	10
■ 授業評価の現状	11
■ 授業フィードバックのあるべき姿	12
■ 授業評価の現状とその問題	13
■ 仮説と役割	
成果物	14
■ "OpenClass." の仕組み	15
■ "OpenClass." [組織側]	20
■ "OpenClass." [ユーザ側]	32
■ 目指すもの	39
制作者がってきたこと	43
■ 準備段階	44
■ 開発段階	52
■ モチベーション	54
レポート	57
■ 所感	58
■ 考察	59
■ 感想	60
■ 謝辞	62
■ 参考文献 / ツール	63

制作者について

■ 三寄遙樹 Misaki Haruki

1999年生まれ。

小・中学校で野球とバドミントンを経験し、高校ではバドミントン部に入部。

幼少期から海外に触れる機会が多く、高校1年でオーストラリアのシドニーで短期のホームステイを経験。

デジタルハリウッド大学入学し、1年次には映像・CGを学び、2年次からC言語やPHPなどのプログラミング言語に触れWebの面白さに気づく。

大学2年次の後期からオーストラリア／ゴールドコーストに留学をし、海外に長期の滞在。新型コロナウィルスの流行により半年間で帰国。

帰国後はWeb制作を独学で学習し、インターンシップにて実務でのWeb制作を学んだ。

Web & UI/UXゼミで初めてデザインやユーザに関わるWebについて学び、さらにWebサイト制作について興味を持つ。

卒業後は独学で学んだ制作ではなく、クライアントと仕事をし、指揮・管理をするディレクターとしてWeb制作会社に入社予定。

■ 矢野 涼 Yano Ryo

小学校から高校まで野球をやっていたこともあり、趣味はお酒を飲みながら野球観戦です。

デジタルハリウッド大学入学後、1年次からC言語やPHPの授業をとり、プログラミングの難しさと面白さに惹かれていきました。

2年次には長年の憧れであった留学を、カナダのトロントで経験。トロントの冬は最低気温-20度と極寒で、髭が凍りついたのはいい思い出です。

3年次からはモバイル開発に興味を持ちSwift言語を学び、作る側だけではなく使う側の視点をもっと深く学んでいきたいと思い、Web & UI/UXゼミに入りました。

卒業後は人材サービス会社のモバイル開発のエンジニア職として入社予定。

概要

■ "OpenClass." とは

"OpenClass." とは、生徒の思う授業に対するリアルな声をオープンにし、授業選択の参考にするためのツールです。

さらに、授業改善をするきっかけを教員に提供します。

リアルな声を公開し、誰もが見れることで、授業選択をスムーズに・授業や教員とのミスマッチを減らし、大学生活をより良いものにしようというサービスです。

■ 詳細

私たちは、“OpenClass.” というサービスを実現するために、以下の制作を行いました。
大学などの教育機関が授業や教員のデータを管理する “OpenClass for University”、
そして、学生や教員が評価された授業を閲覧する “OpenClass for students” です。

これらを使用することで大学における授業の評価を可視化し、学生は授業と本人の相性や
授業の様子を確認するなどの授業選択への活用を目指し、授業を行う教員は授業評価から
授業改善等のきっかけを提供することを目標に制作をしました。

以下、

“OpenClass for University” を [組織側]、

“OpenClass for students” を [ユーザ側] として表記します。

[組織側]

<http://openclass.systems/org/class>

[ユーザ側]

<http://openclass.systems/users/studentsClass>

■ ターゲット

1. 学生

学生は、実際に行われている授業・在籍する教員の評価を見ることができます。これが "OpenClass." の主要な部分です。リアルな評価を見ることができる代わりに、学生には授業に関するアンケートに答えてもらう必要があります。

2. 教員

"OpenClass." の授業の評価は学生だけで無く、教員も閲覧が可能です。教員が実際に学生のリアルな評価を知ることで、授業の良い部分・悪い部分を発見し、改善をするきっかけにしてもらいたいと考えています。

3. 入学希望者

大学入学希望者に向けて授業評価を公開することができるのであれば、大学入学希望者への授業に対するギャップの軽減を行うことができるのではないかと考えています。

4. 大学

ターゲットとして大学を入れたのは、教員が評価を知り授業改善をすれば、結果的に大学として授業全体の質を上げ、さらには大学の質を上げることに繋がると考えたからです。

さらには、入学希望者に向けて授業を公開する際にメールアドレスを取得し、大学に関する情報を送信すると言った、大学としてのマーケティングやビジネスに繋げていけるのではないかと考えています。(※卒業制作時点では機能実装なし。)

■ 競合するサービスとの差別化

"OpenClass." と競合する同様なサービスとして以下のものがあります。

- ・楽単らくだ -<https://www.rakutan-rakuda.com/>
(株式会社 Hushup <https://www.hashup.pro/>)



- ・みんなのキャンパス <https://campus.nikki.ne.jp/>
(楽天グループ株式会社 <https://corp.rakuten.co.jp/>)

Rakuten みんなのキャンパス

これらの評価は有志による書き込みであり、情報量に偏りがあるため、新設授業への懸念があります。「楽単らくだ」の情報は「どう簡単に単位取得をするか」にフォーカスが当てられており、"OpenClass" 本来の目的とは異なります。"OpenClass" では、有志ではなく大学や教員から学生に向けアンケートを取るため、よりリアルで質の高い評価を目指しています。

デジタルハリウッド大学のアプリケーション「デジキャン」と比較するのであれば、アンケートで匿名性を重視していること。リアルな声を書けなければ、結果として授業のフィードバックはできていないと感じ、授業の本質的な改善にはつながっていないと考えています。

経緯

■ なぜ私たちが "OpenClass." を作ったのか

● 背景

私たちが "OpenClass." を制作したのには主に 3 つの背景があります。

1. 大学に入学後、縦の関係が無く、授業に関する事前情報を知ることができなかった。
2. 授業とのミスマッチが原因で、単位を落とすことがあった。
3. シラバスだけでは無く、学生の話を聞きたい。

● 経緯

就職活動を経験した私たちは、「企業と学生のミスマッチを減らすため ...」という言葉をたくさん聞いてきました。実際に企業は学生とのミスマッチを減らすために様々な情報を公開しています。

それを大学というフィールドに置き換えて話すと、授業と学生・教員と学生も同様にミスマッチはない方が良いはずです。

大学の授業環境も、ミスマッチを減らすためにシラバスだけでは無く、「学生からの評価」というものを公開するべきだと私たちは考えました。

大学生活では人それぞれ貴重な時間があります。限られた時間を有効活用するために・より良い授業時間にするために、自分たちを含む学生がミスマッチができるだけ減らし、単位を落としてしまう人を救いたかったのです。

● 前提

○ 大学環境におけるフィードバックシートとは

大学環境におけるフィードバックシート（以下 FS）は本来、学生自身が受講した授業に対し、振り返りをし記憶を整理しアウトプットをするものです。さらには FS を見た教員が学生の理解度などを把握することができます。

○ デジタルハリウッド大学の既存 FS について

デジタルハリウッド大学（以下 DHU）の FS では

- ・授業内容を理解できましたか？（選択必須）
- ・よくわからなかったと記入した方で、授業へのご要望があれば記入してください。（自由記入）
- ・授業の進行スピードについて教えてください。（選択必須）
- ・今回の授業にどのような感想を持ちましたか？（選択必須）
- ・授業の感想をご記入ください（自由記入）

という項目があり、教員へのフィードバック（以下 FB）をすることができます。

※ "OpenClass." では、学生から教員に向けた FB に着目しているため、学生自身の FB の質やメリット・問題点・課題等には触れません。

■ 授業評価の現状

私たちは、現状の DHU の授業環境に以下の問題があると考えています。

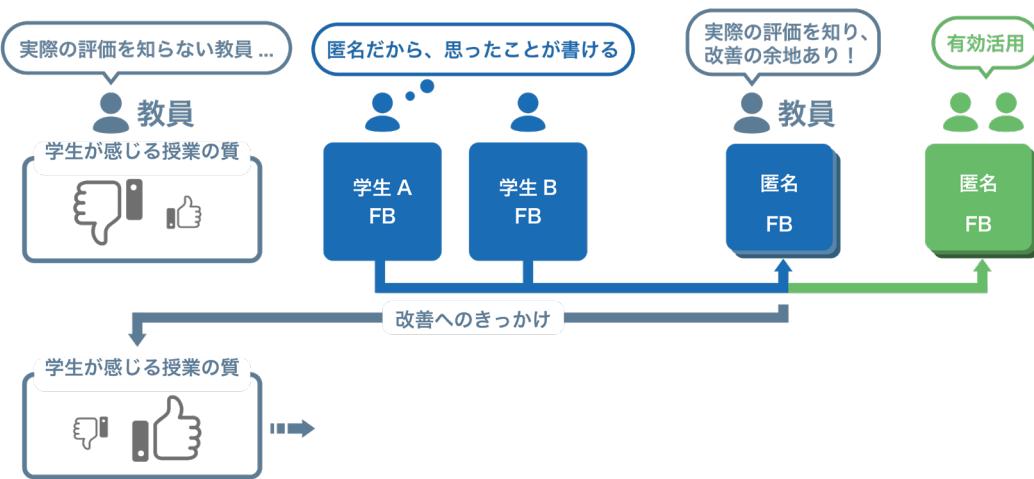
- ・授業後に行われる既存の FS が役に立っていない。
- この問題が全てに関わる課題点であると思います。

■ 授業フィードバックのあるべき姿

学生が授業を受け、リアルな意見を言える環境があるべきだと考えています。

例えば、「今日の授業はペースが早すぎて内容が理解できなかった」「先生が授業の資料を読み上げているだけで面白くなかった」と言った感想や意見をしっかりと発言することができれば、教員はそれを見て改善をする余地があるのであればと思います。

さらに、匿名FBならば、意見しやすい環境なのではないでしょうか。



この授業FBを定期的に行うことができれば授業の改善に繋がり、学生としても質の高い授業を受講できると思います。

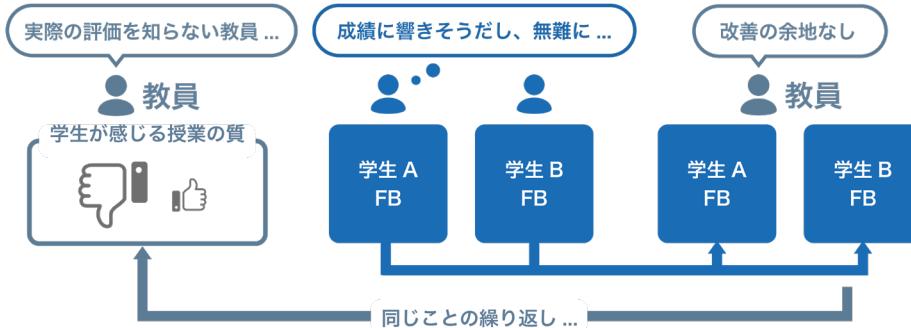
それだけではなく、その感想や意見を学生にも公開することで、どんな授業が行われているのか・どんな教員なのかなど、その授業を選択するにあたり、シラバスだけではわからない情報も収集することができます。

授業のフィードバックを適切に行い、授業の質を可視化する「授業評価」は学生にとっても教員にとっても必要なことのではないかと考えました。

■ 授業評価の現状とその問題

現在 DHU の既存 FS では、どの学生がどの FS を送信したのか中身が全て見えてしまいます。

そのため授業やその教員への意見や感想は書きづらい状態です。学生としては実際にその意見や感想が学生個人の評価に影響されてしまうのではないかと思うからです。



匿名性がないという点で、リアルな意見や感想が書けない

↓

その結果、教員自身が学生の気持ちに気づくことができない

↓

授業・教員の質が向上しない

上記のような問題が繰り返しになるだけだと考えました。

■ 仮説と役割

上記を踏まえ、このような仮説を立てました。

- ・匿名性によりリアルな声を聞くことができる。
- ・リアルな声を適正に公開することで、授業選択での手助けになる。
- ・授業評価を公開することで、授業・教員の質が向上する。

この三つの仮説をもとに、"OpenClass." の制作を始めました。

成果物

■ "OpenClass." の仕組み

● 使用している技術・環境

○ Laravel



『Laravel』は、「MVC」と呼ばれるアーキテクチャを採用している PHP フレームワークです。

この「MVC」とは、

- ・データベースとの連携を担う Model(モデル)
- ・ブラウザへの表示を司る View(ビュー)
- ・指令や指示を司る Controller(コントローラー)

の3つで構成されています。

私たちが Laravel を今回の卒業制作に採用した理由が2点あります。

1. 分業開発がしやすい

上記で示した MVC の役割に関して、矢野が Model, Controller を、三寄が View を担当し役割を分けて作業を進めました。

2. 学習がしやすい

2人がとっかかり易く、継続して学びながら卒業制作を作り出すことが大切だと考え Laravel での制作を選択しました。なぜなら、Laravel は癖のないシンプルなコードで記述でき、Laravel を構成する PHP 自体が「初心者に最適な言語」と言われているからです。以前から独学で PHP のようなシステム的な学習したことのある矢野と、デザインやコーディングの学習を独学で行ってきた三寄にはうってつけの言語でした。

○ HTML



Laravel ではデフォルトで Blade(ブレイド) というテンプレートエンジンが組み込まれています。

View(ビュー) を作成する場合は HTML の雛形を元に画面を作成していきます。

○ CSS/Sass



CSS や Sass は HTML で作成された文書のデザインについて指定を行う際に使用される言語です。

以前は HTML のタグでもスタイルに関する指示を行っていましたが、現在は画像の表示位置や背景や文字の色、大きさなどの指定は CSS や Sass で行うのが一般的となっています。その為、"OpenClass." の作成でも欠かせない言語でした。

○ jQuery



jQuery とは、JavaScript でできることをコンパクトかつ便利にまとめたライブラリ言語です。例えば、クリックしたら画像が入れ替わる、マウスオーバーした要素がアニメーションのように動く、時間によってページの色を変化するなど、JavaScript で書くと時間がかかる作業を jQuery を使えば簡単に記述できます。

○ Chart.js



Chart.js は、データ視覚化用の無料のオープンソース JavaScript ライブラリであり、棒、折れ線、面、円、バブル、レーダー、極、散布の 8 つのグラフタイプをサポートしています。"OpenClass." では、授業評価のグラフを表示する際に五角形のレーダーチャートを使用しました。

○ AWS



AWS とはクラウドコンピューティングを使ったサービスです。

AWS とは Amazon Web Services の略で、Amazon が提供している 100 以上のクラウドクラウドコンピューティングの総称です。

私たちが AWS を利用した理由は、最近のトレンドである点と、何より AWS を使っているサービスが「かっこよくて、イケてる」と思ったからです。

経緯は単純でしたが、使って見ると奥が深く、メリットも多々ありました。

その一つが Laravel と AWS に関するリソースが多かった点です。

○ MySQL

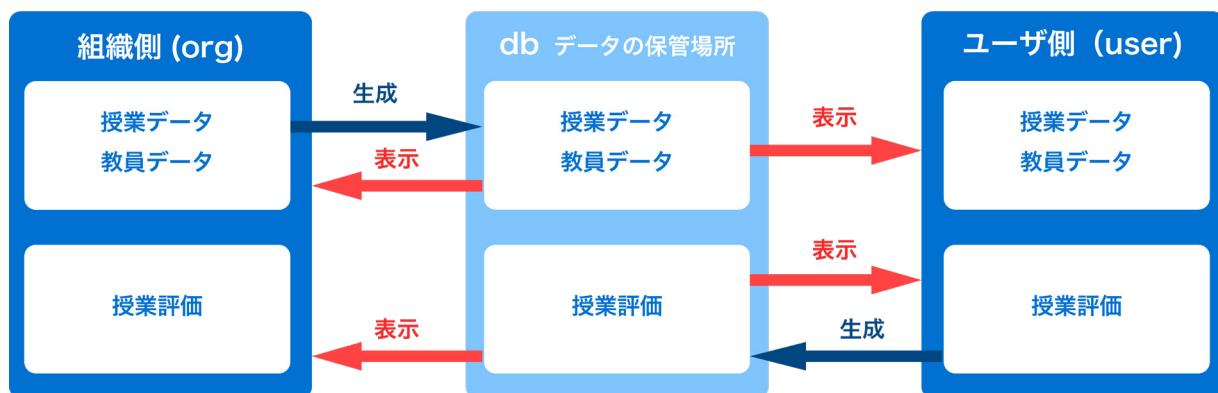


MySQL とは、世界でもっとも利用されている「データベース管理システム」です。

簡単にいようと、データベースとは「複数で共有、利用すること」と「検索、加工すること」を目的に整理されたデータの集まりのことです。

"OpenClass." でも授業や教員の情報をデータベースに入れることにより、それぞれの情報を紐付けて表示させることや、検索することができました。

● 全体構造



"OpenClass." は主に [組織側] と [ユーザ側] のアプリケーションから成り立っています。

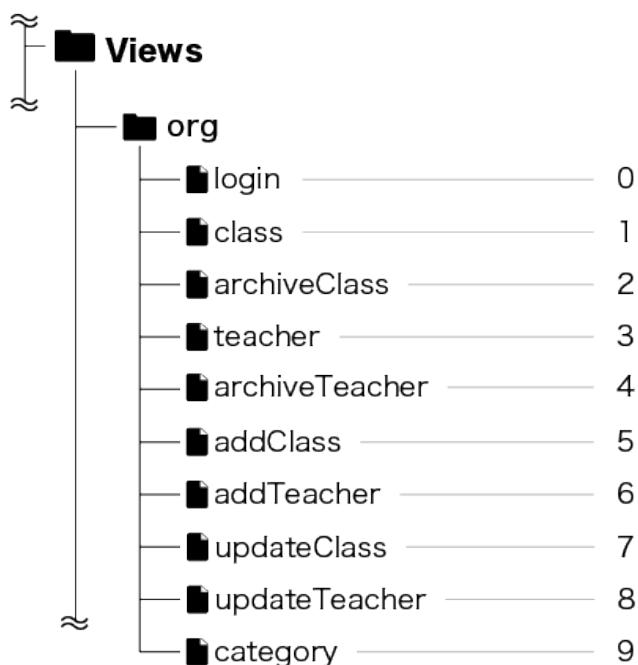
主に授業データ・教員データの管理を行うのが [組織側]、
授業データや教員データをユーザに公開するのが [ユーザ側] です。
全てのデータを管理するのがデータベース（以下 db）の役割になります。

"OpenClass." では [組織側] で生成した授業 / 教員データをデータベースに保管し、
そのデータをユーザ側に表示することで学生や教員に公開するという仕組みになっています。
また逆に、[ユーザ側] で行う授業評価に関するアンケートは、[ユーザ側] から db にデータが生成され、[組織側] や [ユーザ側] に表示されます。

○ ファイル構成

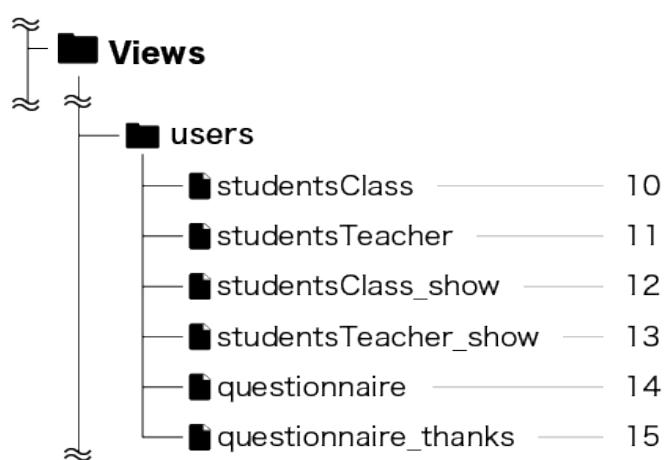
Laravel では View フォルダの配下にあるファイルをブラウザに表示することができます。

[組織側]



※ファイルはわかりやすく順番に並び替えています。

[ユーザ側]



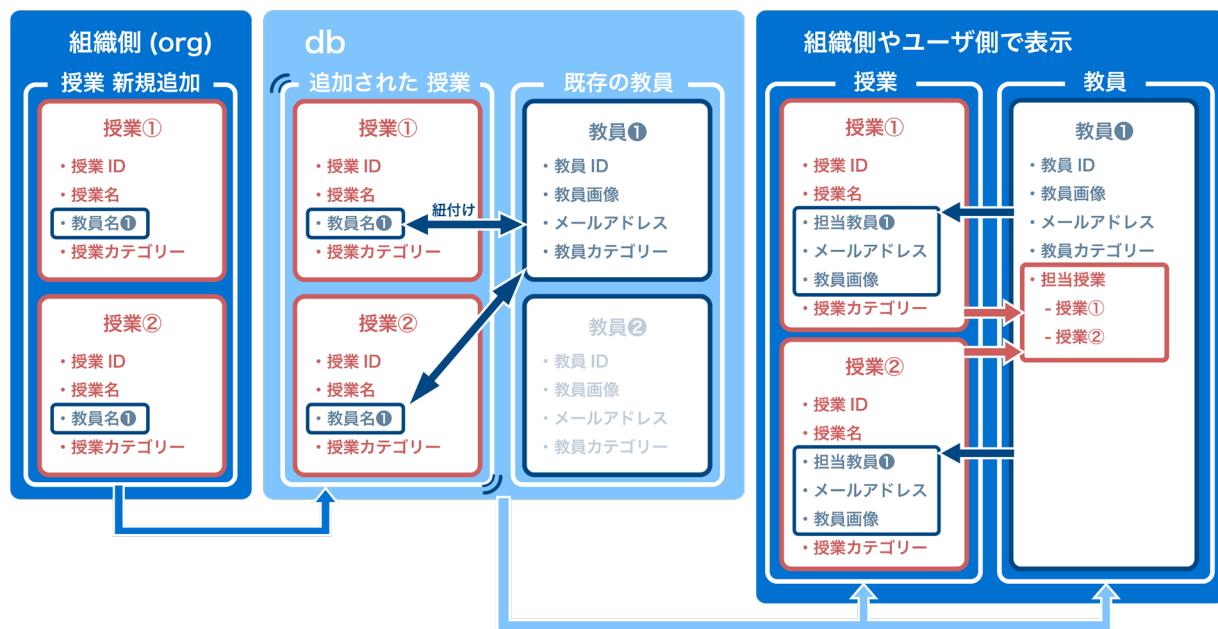
※ファイルはわかりやすく順番に並び替えています。

これらの View フォルダ配下にあるファイルを

[<http://openclass.systems/org/class>] のような形でブラウザに表示します。

■ "OpenClass." [組織側]

● 仕組み



"OpenClass." の [組織側] の仕組みは上図になります。

例として新規授業データを追加します。

図の左にあるように、新規授業追加では授業 ID・授業名・教員名・授業カテゴリを登録します。教員名・授業カテゴリは既存登録されているものから選択をするため、未登録の教員やカテゴリの追加はできません。

以上全てを入力し登録をすると、db にデータ保存されます。(図 : 中央)

db 上では先程追加した新規授業の「教員」を選択することで、その授業データと既存の教員データが紐付けされます。

この紐付けを行い関係性を明確にしておくことで、必要最低限の登録をするだけで表示をする際に多くの情報を表示することが可能になります。(図 : 右)

● [組織側] 全体に共通するデザイン

全てのページに共通する部分はこの下図です。



○ グローバルメニュー

左端の濃い青の部分はグローバルメニューと呼ばれるもので、これから紹介するページを行き来する上で必要なパーツです。

その中には、

- ・授業 / 教員管理を行う [management]
 - ・アーカイブされた授業 / 教員管理を行う [Archives]
 - ・授業 / 教員共通のカテゴリ操作を行う [Category]
 - ・"OpenClass."へのアクセス制限を行う [Access Management]
 - ・授業評価アンケートの管理を行う [Questionnaire Settings]
- という項目があります。

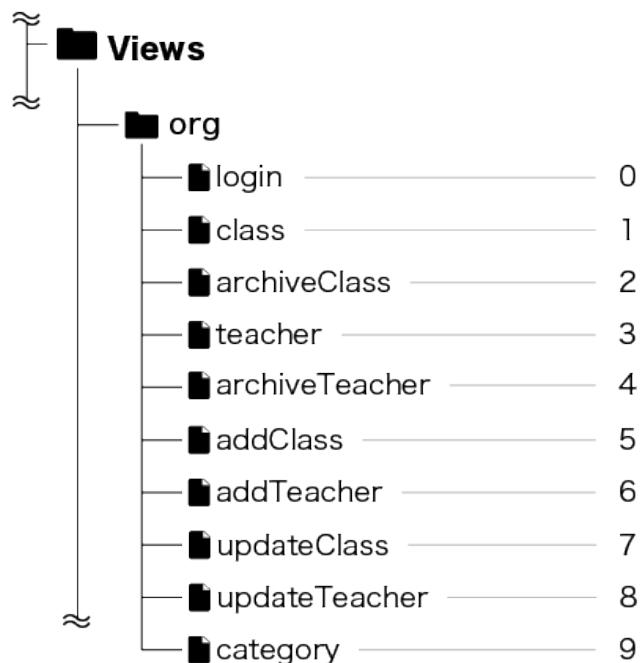
○ ヘッダー

画面上部にある検索バーと、ログアウトボタンです。

アプリケーション内の全ての操作における検索を行う検索バーと、
ログイン・ログアウトを行うために設置されています。

● ページ紹介

[組織側] には下図のファイルがあり、実際にこのファイルがページとして機能しています。



0. openclass/org/login

The screenshot shows the login page of the OpenClass application. On the left, there is a blue sidebar menu with the OpenClass logo and several navigation items: management, Archives, Category, Access Management, and Questionnaire Settings. The main content area has a search bar at the top right. Below it is a login form with fields for E-Mail Address and Password, a 'Remember Me' checkbox, and a 'Login' button. At the top right of the main area, there are 'Login' and 'Register' buttons.

組織側ではまずログインが求められます。

登録をした後にログインをするとトップページに進むことができます。

1. openclass/org/class • 2. archiveClass •

3. teacher • 4. archiveTeacher

これらのページは共通している点が多いため、まとめて紹介します。
"OpenClass." では class・teacher ページがメインのとなります。

○ 共通の機能

ID	Subject	Instructor	Category	action
10012	テクニカルサウンド演習	重松光	演習,文化	⋮
10012	トータルファッショングル	YokoFuchigami	概論,文化,一般	⋮
10013	おもちゃクリエイター概論	安来我楽	概論,一般	⋮
10014	ジェネラルCG演習	KENTA	演習,CG	⋮
10015	インフルエンサー概論	カマタマちゃん	概論,文化,一般	⋮
10016	書道パフォーマンス演習	電彩雲	演習,文化	⋮
10017	謝罪心理概論	坂崎修太	概論,一般	⋮

1. Class / Professor 切り替えボタン

授業管理と教員管理の行き来を可能にする切り替えボタンです。

2. 検索に必要な検索バー / フィルター

全体共通で使われる検索とは違い、授業検索 / 教員検索に特化した検索バーと、カテゴリで検索ができるフィルターです。

3. それぞれのアクションを行うことのできるボタン

授業を複数選択し、まとめてアーカイブや削除のできるボタンです。

4. 一覧表示ヘッダー

以下の簡易表示バーに対応する項目名が記載してあります。
授業管理では、[ID / Subject / Instructor / Category / action]

ID	Subject	Instructor	Category	action
----	---------	------------	----------	--------

教員管理では、[ID / Instructor / Mail / Category / action]

ID	Instructor	Mail	Category	action
----	------------	------	----------	--------

5. 簡易表示バー

一覧表示ヘッダーの項目に対応するものが各授業と各教員で簡易的に表示されます。
action ではマウスオーバーをするとその授業 / 教員単体でアーカイブ、編集、削除を行うことができます。(archive ページではパブリック、編集、削除)
簡易表示バーをクリックするとそれぞれの詳細を見ることができます。

○ class / archiveClass 共通の表示項目

	ID	Subject	Instructor	Category	action
1	10012	テクニカルサウンド演習	重松光	演習,文化	⋮

① ID
10012

② Subject
テクニカルサウンド演習

③ Instructor
重松光

④ Mail
shigematsu@oc.jp

⑤ Category
演習 文化

⑥ Evaluation
3.0

⑦ Number of questionnaires
2

⑧ Number of comments
0

Q1
Q2
Q3
Q4
Q5

簡易表示バーをクリックし、閲覧できる部分が共通です。

1. ID (授業 ID)
2. Subject (授業名)
3. Instructor (教員名)
4. Mail (教員のメールアドレス)
5. Category (授業カテゴリ)
6. Evaluation (総合的な評価と詳細)
7. Number of Questionnaire (総アンケート数)
8. Number of comments (総コメント数)

○ teacher / archiveTeacher 共通

The screenshot shows the 'management' section of the OpenClass application. On the left, a sidebar lists 'management' (selected), 'Archives', 'Category', 'Access Management', and 'Questionnaire Settings'. The main area displays a table of professors with columns for ID, Instructor, Mail, Category, and action. A specific row for '重松光' (Instructor: 重松光, Mail: shigematsu@oc.jp, Category: 演習, 文化) is highlighted. Below the table, a detailed view of this professor is shown with numbered callouts pointing to specific fields:

- 1. ID (教員 ID): 6
- 2. Instructor (教員名): 重松光
- 3. Image (教員画像): A placeholder image of a person.
- 4. Mail (教員のメールアドレス): shigematsu@oc.jp
- 5. Category (教員カテゴリー): 演習, 文化
- 6. Subject in charge (担当授業): テクニカルサウンド演習

At the bottom of the page, there are navigation links for 'Logout', 'Classes', 'Professors', 'Search', 'Filter', 'Archive', 'Delete', and a 'Logout' button.

簡易表示バーをクリックし、閲覧できる部分が共通です。

1. ID (教員 ID)
2. Instructor (教員名)
3. Image (教員画像)
4. Mail (教員のメールアドレス)
5. Category (教員カテゴリー)
6. Subject in charge (担当授業)

○ポイント

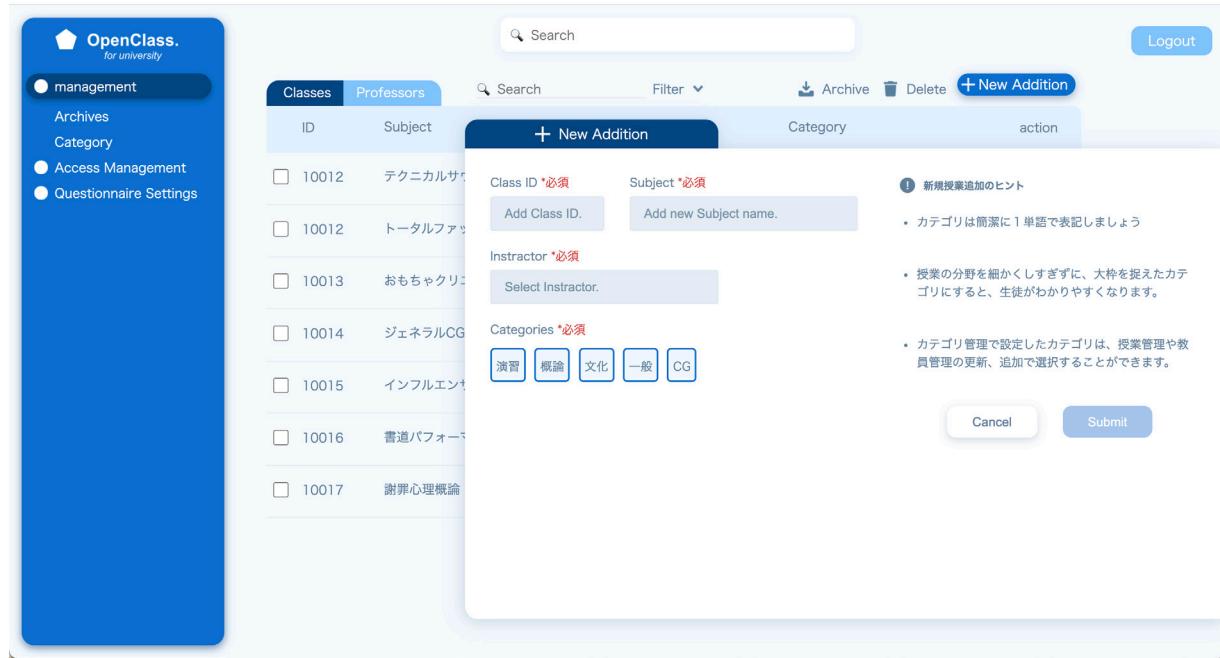
特に class · teacher ページは [組織側] にとって一番重要なページと言っても過言ではないページだったので徹底的に作り込みをしました。

気持ちよく動くアニメーションにも力を入れ、action ボタンではマウスをかざすとユーザが取れる行動を表示し、使う頻度の少ないボタンを隠すことでシンプルなデザインにしました。

使用頻度も少なく目立った機能ではないのですが、実装にはかなり時間がかかりました。

5. openclass/org/addClass • 6. addTeacher

addClass・addTeacher では授業・教員の新規追加をすることができます。
class・teacher ページ上で [+ New Addition] ボタンを押すと、画面右側からぬるっと
新規追加のボードが流れています。



上図の新規授業追加では、

入力項目：class ID（授業 ID）・Subject（授業名）

選択項目：Instructor（教員名）・Categories（カテゴリ）

全てで 4 項目を登録します。

成果物 / ■ "OpenClass." [組織側] / ● ページ紹介 / 5・6

The screenshot shows the 'Professors' management section of the OpenClass application. On the left, a sidebar lists 'management' options: Archives, Category, Access Management, and Questionnaire Settings. The main area displays a list of existing professors with columns for ID and Instructor. A modal window titled '+ New Addition' is open, prompting for new professor details. Required fields include 'Professor ID *必須' (with placeholder 'Add Prof ID.'), 'Professor name *必須' (with placeholder 'Add new Professor name.'), and 'Mail *必須' (with placeholder 'Add new Mail address.'). An 'Image' field contains a placeholder image, with an 'Upload image' button below it. A 'Categories *必須' section includes five buttons: 演習 (Exercises), 概論 (General Theory), 文化 (Culture), 一般 (General), and CG. A note on the right provides hints: '新規教員追加のヒント' (Hint for adding a new teacher), 'カテゴリは簡潔に1単語で表記しましょう' (Please represent categories简洁ly in one word), '授業の分野を細かくしきぎずに、大枠を捉えたカテゴリにすると、生徒がわかりやすくなります。' (When specifying course fields, do not be too detailed; instead, capture the broad categories), and 'カテゴリ管理で設定したカテゴリは、授業管理や教員管理の更新、追加で選択することができます。' (Categories set in category management can be selected for updates and additions in lesson and teacher management).

上図の新規教員追加では、

入力項目 : Professor ID (教員 ID) · Professor name (教員名) · Mail (メールアドレス)

選択項目 : Image (教員画像) · Categories (カテゴリ)

全てで 5 項目を登録します。

上図のような入力していない段階では、[Submit] ボタン (送信ボタン) が機能しない状態ですが、

下図のように必須項目を全てを入力している段階になると、[Submit] ボタンが機能します。

The screenshot shows the 'Classes' management section of the OpenClass application. On the left, a sidebar lists 'management' options: Archives, Category, Access Management, and Questionnaire Settings. The main area displays a list of existing classes with columns for ID and Subject. A modal window titled '+ New Addition' is open, prompting for new class details. Required fields include 'Class ID *OK' (with placeholder '10018') and 'Subject *OK' (with placeholder 'デジタルコミュニケーション'). An 'Instructor *OK' field contains the placeholder '矢野涼'. A 'Categories *OK' section includes five buttons: 演習 (Exercises), 概論 (General Theory), 文化 (Culture), 一般 (General), and CG. A note on the right provides hints: '新規授業追加のヒント' (Hint for adding a new class), 'カテゴリは簡潔に1単語で表記しましょう' (Please represent categories简洁ly in one word), '授業の分野を細かくしきぎずに、大枠を捉えたカテゴリにすると、生徒がわかりやすくなります。' (When specifying course fields, do not be too detailed; instead, capture the broad categories), and 'カテゴリ管理で設定したカテゴリは、授業管理や教員管理の更新、追加で選択することができます。' (Categories set in category management can be selected for updates and additions in lesson and teacher management).

登録する内容は 4~5 項目ですが、実際に class · teacher ページで表示される情報としては 6~8 項目に変化します。仕組みでも記載したように、あらゆるデータを紐付けしているため、必要な情報をあらゆる場所から引っ張ってこれるようになっています。

○ ポイント

このページで頑張ったのは、リアルタイムのバリデーション（入力チェック）です。技術的にjQueryというツールに頼るしかなかったのですが、非常に大変でした。特に、全ての必須項目を入力すると[Submit]できるようになるという部分では非常に難航しました。多くの検証と実験を重ねて完成することができました。

7. openclass/org/updateClass · 8. openclass/org/updateTeacher

updateClass · updateTeacher では授業・教員の変更、修正をすることができます。class · teacher ページの簡易表示バーの [action] ボタンの [編集] ボタン（鉛筆マーク）を押すことで授業情報や教員情報を編集することができます。それぞれ元データが入力された状態で編集ページに遷移します。機能としては addClass · addTeacher と変わりません。

The screenshot shows the 'Edit Information' form for a class. The left sidebar has a 'management' section with 'Archives', 'Category', 'Access Management', and 'Questionnaire Settings'. The main form fields are:

- Class ID *OK: 10012
- Subject *OK: テクニカルサウンド演習
- Instructor *OK: 重松光
- Categories *OK:
 - 演習
 - 概論
 - 文化
 - 一般
 - CG

On the right, there is a sidebar with tips for adding classes:

- 授業追加のヒント
- カテゴリは簡潔に1単語で表記しましょう
- 授業の分野を細かくしきぎざに、大枠を捉えたカテゴリにすると、生徒がわかりやすくなります。
- カテゴリ管理で設定したカテゴリは、授業管理や教員管理の更新、追加で選択することができます。

At the bottom are 'Cancel' and 'Submit' buttons.

上図：updateTeacher

成果物 / ■ "OpenClass." [組織側]/ ● ページ紹介 /7・8

The screenshot shows the 'Edit Information' page for a teacher. On the left, there's a sidebar with the 'OpenClass. for university' logo and navigation links: 'management' (selected), 'Archives', 'Category', 'Access Management', and 'Questionnaire Settings'. The main area has a search bar and a 'Logout' button. The 'Edit Information' form contains fields for 'Professor ID *OK' (1011), 'Professor name *OK' (重松光), 'Mail *OK' (shigematsu@oc.jp), and an 'Image' section with a placeholder profile picture and an 'Upload image' button. Below these are 'Categories *OK' buttons for 演習, 概論, 文化, 一般, and CG. To the right of the form is a 'HINT' section with three bullet points: 1. 教員追加のヒント (Hint for adding staff). 2. カテゴリは簡潔に1単語で表記しましょう (Category should be represented in one word). 3. 授業の分野を細かくしそぎずに、大枠を捉えたカテゴリにすると、生徒がわかりやすくなります。 (When specifying subject fields, it's better to capture the broad categories rather than being too specific). At the bottom right are 'Cancel' and 'Submit' buttons.

上図：updateTeacher

○ ポイント

本来は、class・teacher ページから addClass・addTeacher ページのように横からスライドインしてくるようにしたかったのですが、私たちの技術不足でそれを実現することができませんでした。

9. openclass/org/category

category ページでは授業や教員に関わるカテゴリを追加・削除することができます。

The screenshot shows the 'Category' management page. On the left, there's a sidebar with navigation links: 'management', 'Archives', 'Category' (which is highlighted), 'Access Management', and 'Questionnaire Settings'. The main area has a search bar at the top right. Below it is a section titled 'Enter Categories' with a plus sign (+) button. A list of existing categories is shown in blue boxes: '演習', '概論', '文化', '一般', and 'CG'. Below this is a button labeled 'Select categories and delete.' To the right, there's a note about category hints and a list of tips:

- カテゴリのヒント
- カテゴリは簡単に1単語で表記しましょう
- 授業の分野を細かくしそぎずに、大枠を捉えたカテゴリにすると、生徒がわかりやすくなります。
- カテゴリ管理で設定したカテゴリは、授業管理や教員管理の更新、追加で選択することができます。

カテゴリを追加すると [+] ボタンが青く光り、追加をすることができます。下図

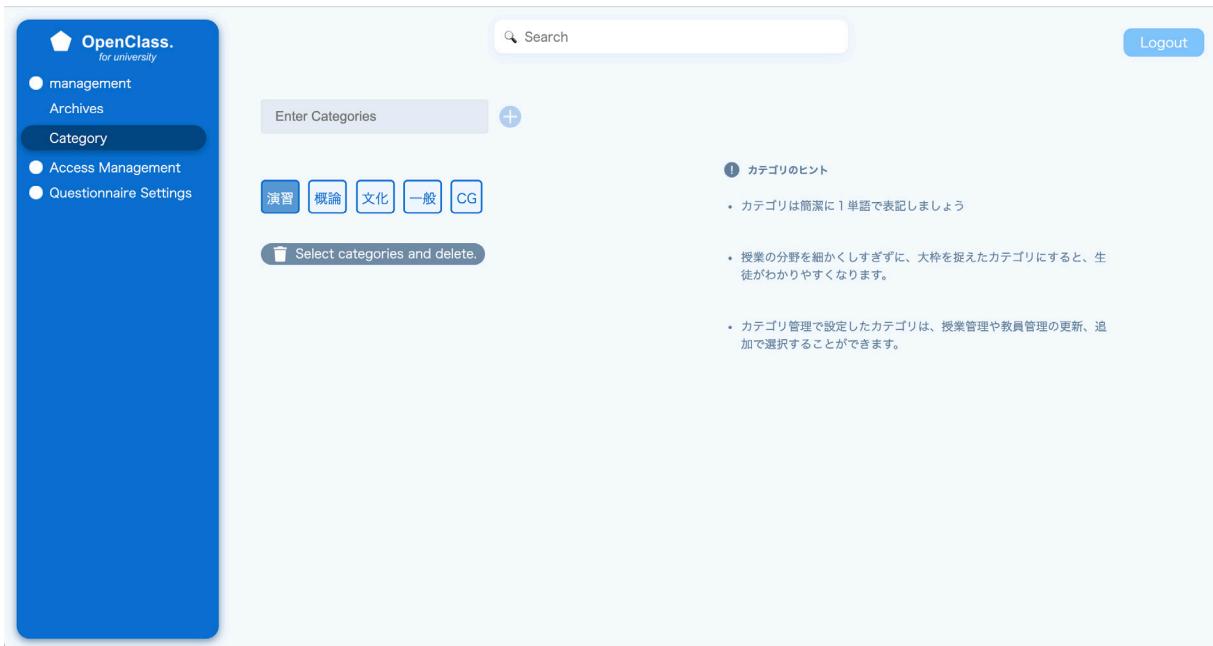
This screenshot shows the same 'Category' management page after a new category has been added. The 'Enter Categories' input field now contains the word 'デザイン' (Design). The '+ ' button next to the input field is now highlighted in blue, indicating it is active. The rest of the interface, including the sidebar, existing categories, and tips, remains the same as the previous screenshot.

成果物 / ■ "OpenClass." [組織側] / ● ページ紹介

ただし、重複するカテゴリを入力した場合は、注意書きとともに [+] ボタンが機能しなくなります。下図



さらに、既存のカテゴリを選択することで [Select categories and delete.] ボタンが機能し、クリックすることで削除することができます。下図

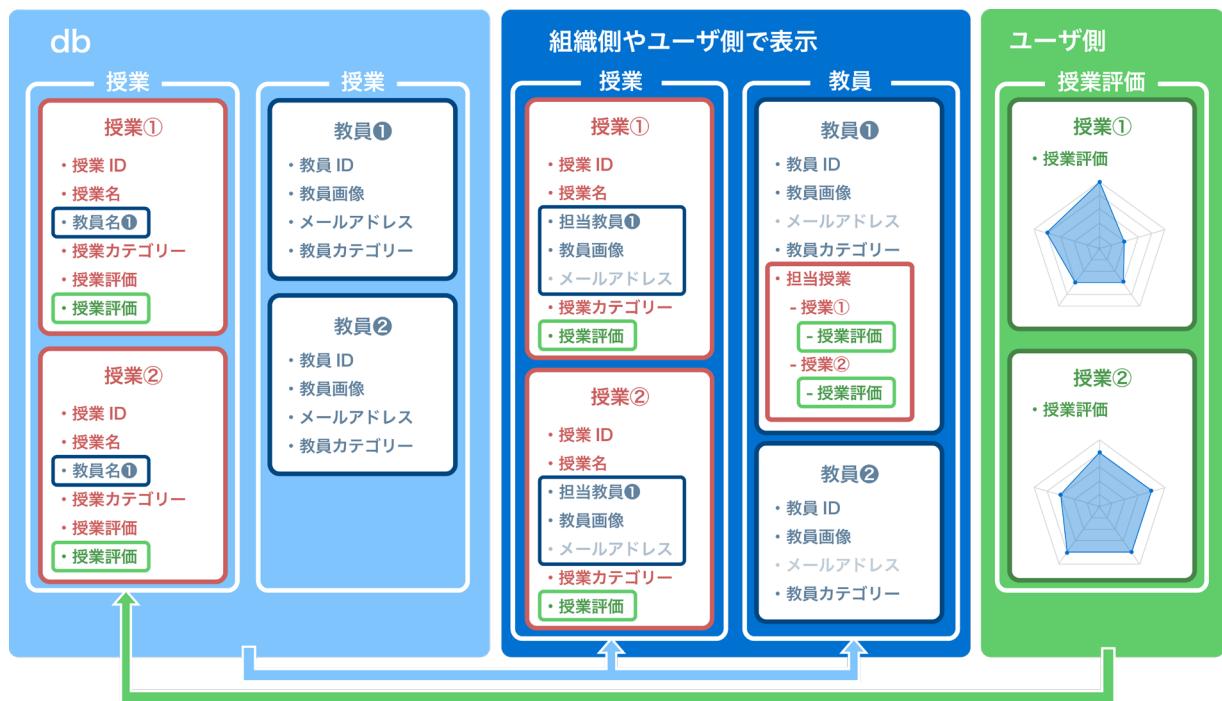


○ ポイント

この機能に関しても、授業・教員の新規追加と同じ機能が使われ、入力・選択したタイミングで不備があればボタンが機能しないなどの工夫がされています。

■ "OpenClass." [ユーザ側]

● 仕組み

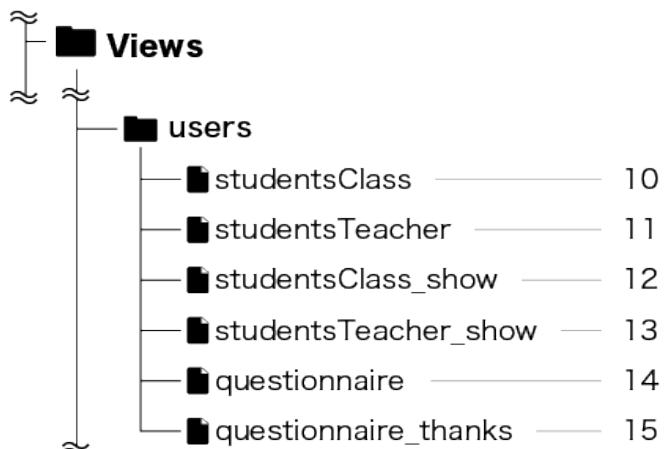


基本的なデータの動き自体は [組織側] と変わりませんが、[ユーザ側] には授業アンケートを行うページ (openclass/users/questionnaire) が存在します。

上図右の [ユーザ側] 授業評価を学生に登録してもらいます。登録されたデータはすべてそれぞれの授業に紐付けされ [db] に保存され、[ユーザ側] や [組織側] に表示するという流れです。

● ページ紹介

[ユーザ側] には下図のファイルがあり、実際にこのファイルがページとして機能しています。



10. openclass/users/studentsClass •

11. openclass/users/studentsTeacher

studentsClass ページでは授業に関する情報が表示され、
studentsTeacher ページでは教員に関する情報が表示されます。



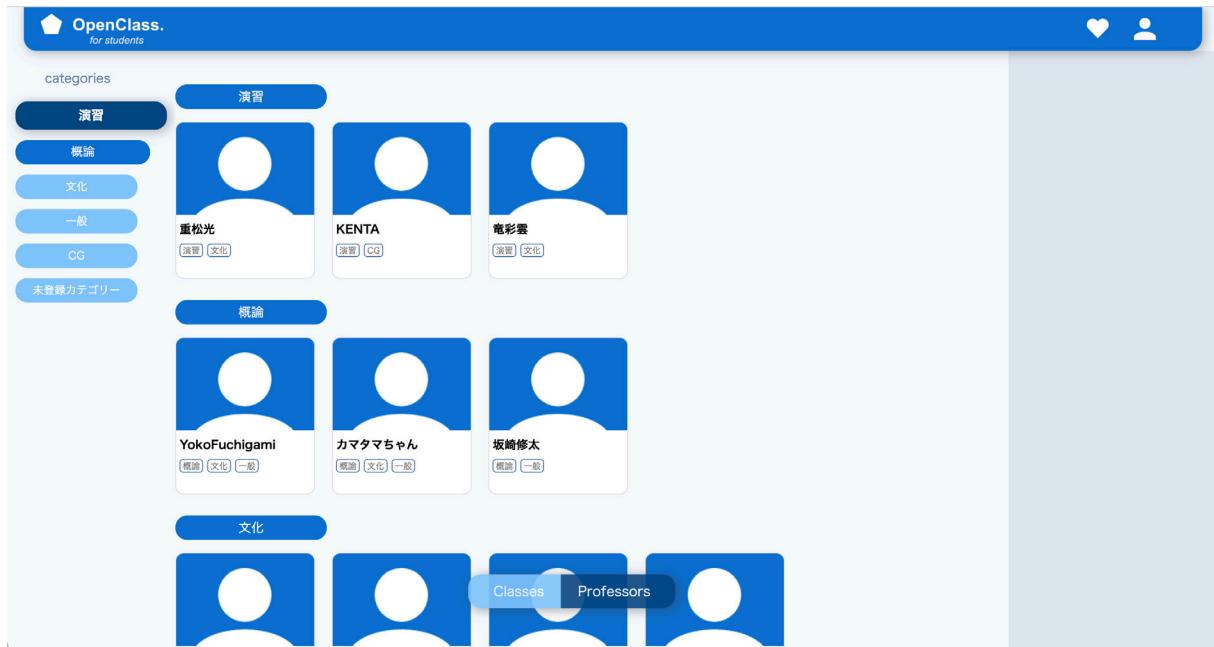
上図：studentsClass

各ページの授業や教員はカテゴリ別に分けられ「カード型」で表示されます。

studentsClass ページでは

- ・教員画像
- ・評価
- ・授業名
- ・授業カテゴリ

がカード内に表示されます。



上図 : studentsTeacher

studentsTeacher ページでは

- ・教員画像
- ・教員名
- ・教員力テゴリ

がカード内に表示されます。

画面左側のカテゴリボタンをクリックすると、そのカテゴリが表示してある場所までスクロールします。

画面下の Classes/Professors ボタンは授業一覧と教員一覧ページの切り替えができます。

○ ポイント

本来実現したかったものがあります。

1. ログイン・ログアウト
2. いいね機能
3. ランキング

個人のアカウントを作るとこで、「いいね」などの保存機能の追加をしたかったです。

また、画面右側の色が変わっている部分には授業評価のランキングを表示したいと考えていました。

そのほかにも、学生がどの授業を受けているのかを把握することができれば、個人のアカウントを登録するタイミングで将来就きたい職業を選択し、「〇〇 (Web デザイナー) になりたい人は、この授業を受講しています」などのおすすめを表示するコンテンツを実現したかったのですが、アンケートを匿名にしたことなど、矛盾点が生じてしまい実現することができませんでした。

12. openclass/users/studentsClass_show •

13. openclass/users/studentsTeacher_show

studentsClass_show ページでは授業の詳細を表示し、
studentsTeacher_show ページでは教員の詳細を表示します。



The screenshot shows the 'studentsClass_show' page. At the top, there's a blue header bar with the 'OpenClass.' logo and navigation icons. Below the header, the class title '3.0 テクニカルサウンド演習' is displayed, along with category tags '演習' and '文化'. The teacher's name '担当教員: 重松光' is listed. A large '授業評価: 3.0' is shown above a 5-point rating graph. The graph has five points labeled Q1 through Q5 around its perimeter. To the right of the graph is a 'Question' section with five numbered questions. Below the graph and question section are two comment sections, each with a 'コメント' heading and a note '評価がありません。'. At the bottom, there are tabs for 'Classes' (which is selected) and 'Professors'.

上図 studentsClass_show では、

- ・教員画像
- ・授業名
- ・授業カテゴリ
- ・担当教員名
- ・授業評価
 - グラフ
 - 評価項目
 - コメント 3つ

が表示されます。

OpenClass.
for students

重松光

教員の一言

こんにちは私の名前は重松光です。これから授業全力で楽しめよう！

担当科目

3.0 テクニカルサウンド演習

Classes Professors

上図 studentsTeacher_show では、

- ・教員画像
 - ・教員名
 - ・教員からのメッセージ
 - ・担当科目
- が表示されます。

○ ポイント

教員項目に教員からの一言を設置した理由は、教員の印象を生徒に与えるためです。
どんな人柄なのかは写真だけではわからないことが多いので、少しでも教員についてどのような人なのかを提供するために教員からの一言を設置しました。

14. openclass/users/questionnaire •

15. openclass/users/questionnaire_thanks

questionnaire ページは授業評価のアンケートを行うページです。

The screenshot shows the 'questionnaire_thanks' page from OpenClass. At the top, there's a header with the OpenClass logo and the text 'OpenClass. for students'. Below the header is a user profile placeholder with the name 'テクニカルサウンド演習' and '教員: 重松光'. A message reads: '授業お疲れ様でした！ 97秒くらいのアンケートで、完全匿名です！ 答は全自由項目ですので、答えられるところだけ答えてください！ ではよろしくお願いします！'.

The page contains seven survey questions (Q1 to Q7) each with five response options (thumbs up, thumbs down, etc.) and a text input field for Q6 and Q7. Q8 has a text input field.

Q1, この授業を誰かに勧めたいと思いますか？

Q2, 「授業の内容」と「シラバスの内容」は概ね一致していましたか？

Q3, この授業を行なっている教員と授業をするのは楽しいですか？

Q4, この授業は簡単でしたか？

Q5, この授業のベースは他の授業のベースに比べてちょうどよかったです？

Q6, あえていうなら、
例：回線が悪い _____ ところを直せば、
例：授業がもっとスムーズになる _____ と思う...。

Q7, 例：授業のスムーズさ _____ は他の授業よりも最高！

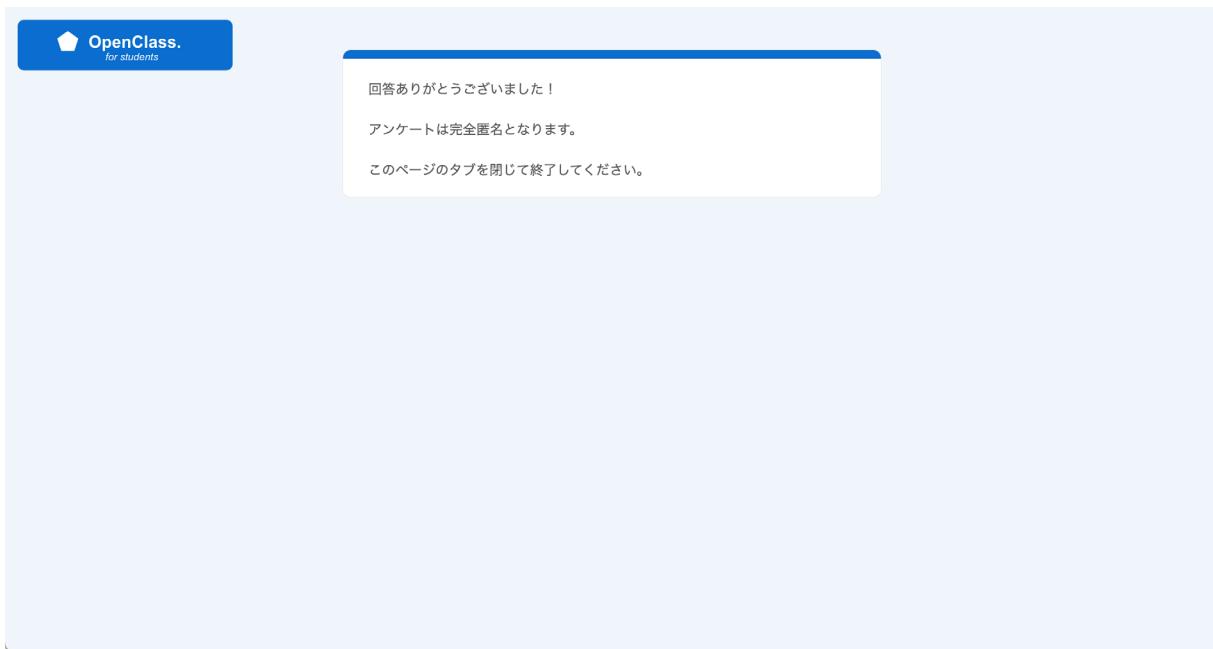
Q8, 何か 重松光 にコメントがあれば！ _____

At the bottom right is a circular timer showing 'now 77s'. In the center is a blue 'Submit' button.

アンケートは全て匿名で行うためログイン等は必要ありません。

評価としては7項目あり、全て自由項目なので答えることのできる部分だけ回答が可能です。

前ページの [Submit] ボタンを押すと、下図の questionnaire_thanks ページに遷移し終了します。



○ ポイント

匿名でログイン等する必要がないため、URL を知っていれば何度でもアンケートに回答できるという点が懸念点になっています。

画面右下の時間が減っていくタイマーをつけた理由としては、だらだら何かを記入するよりもある程度時間を区切ってサクッとアンケート記入をしてもらいたいと考えたからです。

ただ、時間が0秒になっても強制的に終わることはありません。

■ 目指すもの

私たちは当初、デジタルハリウッド大学という教育機関で "OpenClass." を利用してもらうことを目標に制作を開始しました。

実際にそのレベルまで到達することは叶いませんでした。

私たちがこれからサービスをより良くしていくために目指していくものをお紹介していきます。

1. [組織側] ログイン

- ・現状

現状、[組織側] は Register ボタンから登録すれば、誰もがログイン可能な状態になっています。

- ・目指すもの

どのメールアドレスでログインできるのかを [組織側] の Access Management から操作できるようにしたいと思っています。

2. [組織側] アンケート設定

- ・現状

現状、アンケートの内容が変更出来ない仕組みになっています。

- ・目指すもの

アンケート内容は柔軟に変更できるようにしたいと思っています。Questionnaire Settings から質問項目を変更できるようにします。

さらに、アンケートを実施する時期を設定すると、授業評価ページの URL が教員に自動的に送信されるようにし、教員が授業の最後にアンケート実施を促すという流れにしたいです。

3. [組織側] 授業・教員の修正 / 変更

- ・現状

現状、修正 / 変更をすると、ページが遷移してしまいます。

- ・目指すもの

ページ遷移をせずに、授業・教員追加のように画面端からスライドインするような形にしたいと思っています。

4. [組織側]一斉アーカイブ / 削除

- ・現状

現状、授業ページや教員ページには ID の横にチェックボックスがありますが、何も機能しない状態にあります。

- ・目指すもの

必要なものにチェックをし、画面上の Archive/Delete を押すと一斉にアーカイブできる、削除できるという形にしたいと思っています。

5. [組織側]CSV ファイル読み込み

- ・現状

新規の授業や教員を一つ一つ手作業で追加していく状態で、多くの授業や教員を追加するのにかなりの時間がかかります。

- ・目指すもの

CSV ファイルを読み込む機能を追加したいです。実際に CSV で読み込む機能自体は可能であり、開発段階ではそのように追加していたこともあったのですが、現状ではそのように追加するボタンやページがないので、そのような機能を追加したいです。

6. [組織側]レスポンシブ対応

- ・現状

現状、画面サイズがスマホやタブレットに対応していません。

- ・目指すもの

スマホやタブレットで管理することを想定していないというのが作らなかった理由なのですが、タブレットの画面サイズには対応させようと思っています。

7. [ユーザ側]検索機能

- ・現状

現状はカテゴリのみの分類になっています。

- ・目指すもの

カテゴリだけでなく、詳細な絞り込み機能を搭載しユーザが検索しやすいような機能を搭載したいです。

8. [ユーザ側] ユーザ登録

- ・現状

ユーザ登録なし

- ・目指すもの

ユーザ登録機能を追加することで受けられる恩恵は非常に多いと思います。代表とすれば「いいね」「お気に入り」のような保存しておくといった機能です。

ユーザ登録をする際に、「将来どんな職業につきたいのか」を知ることができれば、その人が受けていた授業や、お気に入りにした授業が何かを判別し、例えば他のユーザに「CGクリエイターになりたい人はこんな授業をお気に入り登録しています。」といったおすすめなどを提示することができるのではないかと思っています。

9. [ユーザ側] お気に入り機能

- ・現状

お気に入り機能なし

- ・目指すもの

お気に入り機能はユーザ登録を実現すると可能になる機能です。

ユーザがどのような授業に興味があるのかを知ることができれば、[ユーザ側] のコンテンツが非常に面白くなっていくのではないかと思っています。

10. [ユーザ側] ランキング

- ・現状

機能なし

- ・目指すもの

授業のランキングをつけると、教員のモチベーションにもつながり、学生の興味関心が強くなると思います。

● まとめ

[組織側] では機能面がまだまだ足りていないと感じており、[組織側] を使うユーザにもっと使い勝手が良いと感じてもらうように機能拡充に勤めたいです。

また、[ユーザ側] はコンテンツが少ないと感じており、コンテンツ拡充をすることでさらに面白いサービスにできると自負しています。

現在出来上がっている "OpenClass." は必要最低限の機能が備え付けてあるベータ版のような状態だと思っています。

もちろん、私たちは卒業制作として技術力・経験・時間が少ない中で精一杯に取り組んできました。

だからこそコンテンツ量・機能・ユーザ体験など、まだまだ満足できずにいるのが現状です。

これからさらに良いサービスにできるように頑張りたいと思います。

制作者が行ってきたこと

制作者が行ってきたこと /

ここからは "OpenClass." を作るにあたり、三崎・矢野がどのようなことをしてきたのかを記載していきます。

■ 準備段階

● 目的・役割の明確化

当時、卒業制作では何をやりたいのかをイメージすることが出来ていたため、かなり早い段階で目的や "OpenClass." の役割を明確化することができました。

特に「誰かのために何かを解決したい」という気持ちが強く、共通の思いを持った2人だったので、目的にあたる「誰のため・何のためにサービスを作るのか」や、役割である「どんなサービスにしていくか」という部分はすぐに決めることができました。

● 似たサービスのリサーチ

どんなサービスを作りたいのかが決まった時点で、同様のサービスがあるのかを探しました。

実際に似ているサービスがいくつかあったのですが、幸いと言っていいのか私たちが目指しているサービスと全く同様のものはありませんでした。

サービス制作において参考にできるものは参考にし、差別化できる部分はしっかりと差別化するように心掛け、私たちが目指していく部分がわかったような気がしました。

● サービスの方向性の明確化

同様なサービスを確認し、どんな部分が "OpenClass." の強みなのか・実現すべきことは何なのか・ターゲットが誰なのか・目的は何かを改めて再確認をし、どの方向性に向かっていくのかを吟味しました。

目的・役割の明確化がしっかりとできていると、サービスの方向性は比較的簡単に進んでいくのかと思います。同様なサービスを見て、差別化を測れるところを強みにしていくという点はとても面白かったです。

● サービスの名前検討

"OpenClass." は「リアルな授業評価を公開する」を目的に制作をしました。

さらに、授業評価のみならず授業自体の雰囲気や授業での教員の雰囲気も公開する、

まさに「授業をオープンにする」という意味を込めて命名しました。

● サービス課題の洗い出し

このパートはかなりの時間がかかりました。

サービスを使ってもらうにあたり、どんな問題があるのか、どのようにしたら使ってもらえるのか、授業評価はどのように収集するのか、どうしたら授業評価を答えてくれるのか、学生へのアプローチの仕方や、大学に簡単に使ってもらうためにどんな工夫が必要なのか、などたくさんの課題が出てきました。

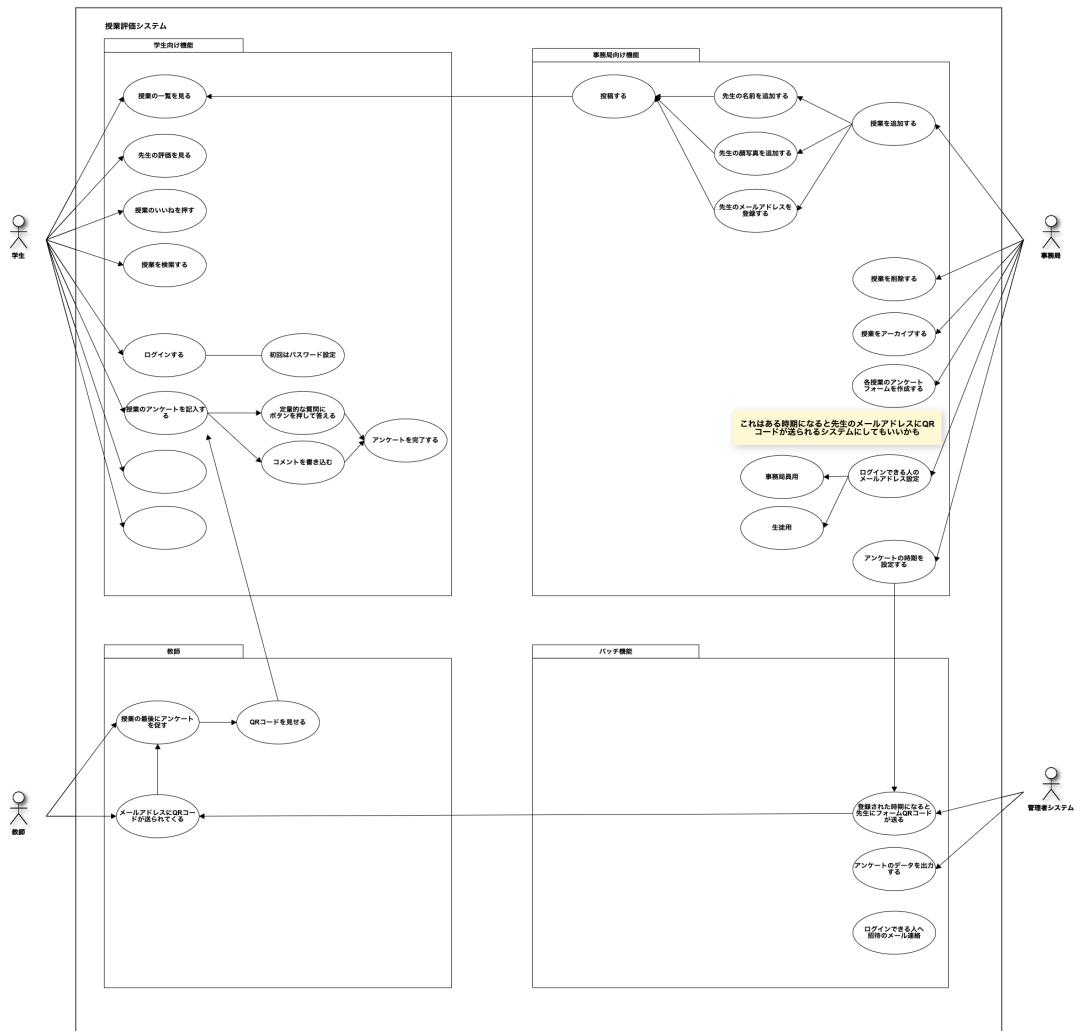
解決方法を一つ一つ考えているうちにさらなる課題が浮上し、なかなか解決方法が決まらないという事態に発展しました。

● 必要なコンテンツ選定

目的や課題を解決するためにどのようなコンテンツが必要なのかを考えていきました。

私たちの制作は、主に学生・教員が見るコンテンツ [ユーザ側] と大学のスタッフが見るコンテンツ [組織側] の 2 つを制作しなければならなかったので、それぞれ何が必要なのかなど、お互いのコンテンツの関係を考えながらコンテンツ選定をしていました。

● ユースケース図作成



上図：当時作成したユースケース図

必要なコンテンツが大まかに決まった後は、ユースケース図を作成しました。

ユースケース図は、システムで何ができるかを「ユーザー目線で」表現する図解術です。「システムを利用する人の目線で」具体的にシステムを利用する場面を想定して、視覚的に図示することがユースケース図を使う目的です。

参考・引用：cacoo ユースケース図とは？書き方や注意点を初心者でも分かるように解説

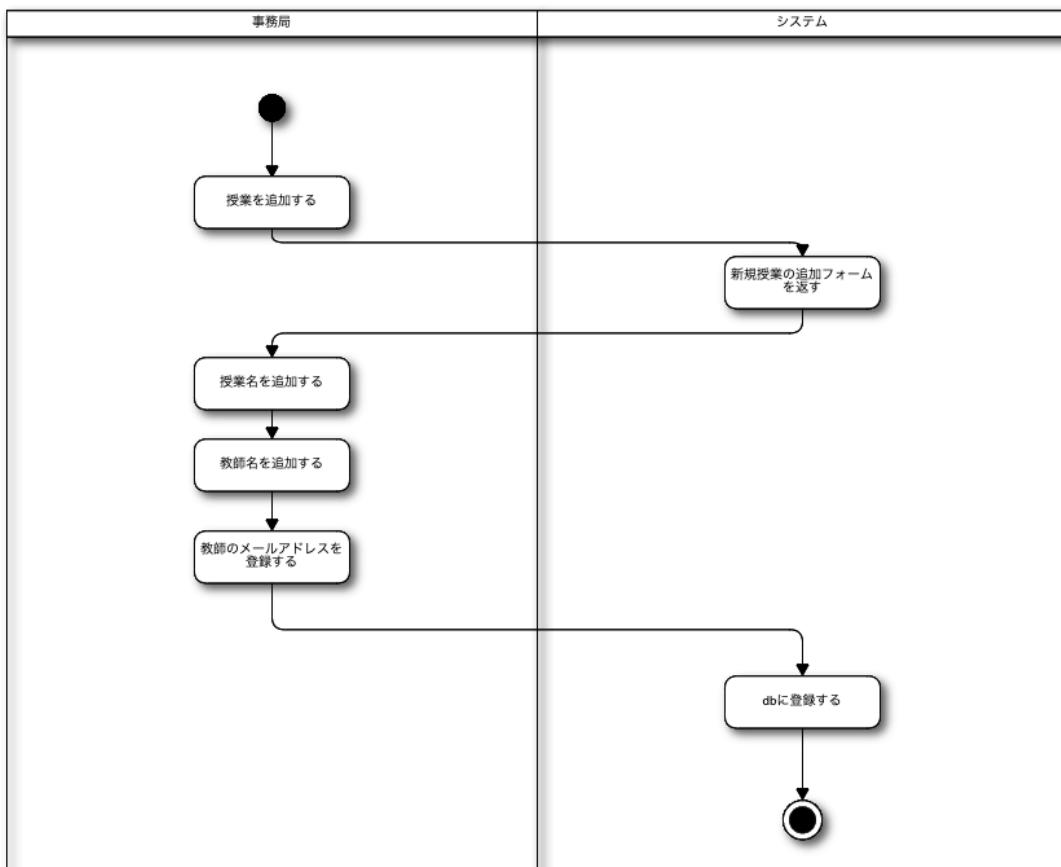
URL : [https://cacoo.com/ja/blog/what-is-use-case-diagram/]

これを使うことで、ユーザーが実際に使った時にシステムがどのように動くのかを簡単に見ることができ、[ユーザ側]と[組織側]のシステムの相関関係を図で表すことができるので非常に便利です。

私たちは簡単なものしか作ることができないのですが、「ユースケース図」を含め、この後紹介する「アクティビティ図」「画面一覧表」「画面遷移図」「テーブル設計図」は作れると開発段階で作業が捲ると思います。

● アクティビティ図作成

[授業の追加]



上図：当時作成した授業追加のアクティビティ図

具体的なサービスの使い方の流れが決まった後は、アクティビティ図を作成しました。アクティビティ図とは UML（統一モデリング言語）の一種で「システム実行の流れと条件分岐」を図解したものです。具体的には、ある作業の開始から終了までの機能を、実行される順序どおりに記述します。

参考・引用：[**cacoo** アクティビティ図（フローチャート）の作り方・活用法を初心者向けに解説](#)

URL : [<https://cacoo.com/ja/blog/what-is-activity-diagram/>]

例えば、[組織側] で授業を追加した場合に、どのようなシステムが実行するのか、さらにそれを受け [組織側]・[ユーザ側] で何が起こるのかの流れを図で表現するものです。これを作ることで開発の段階でやるべきことが整理されるので、あとで悩まなくて済むので非常に便利です。

● 画面一覧表作成

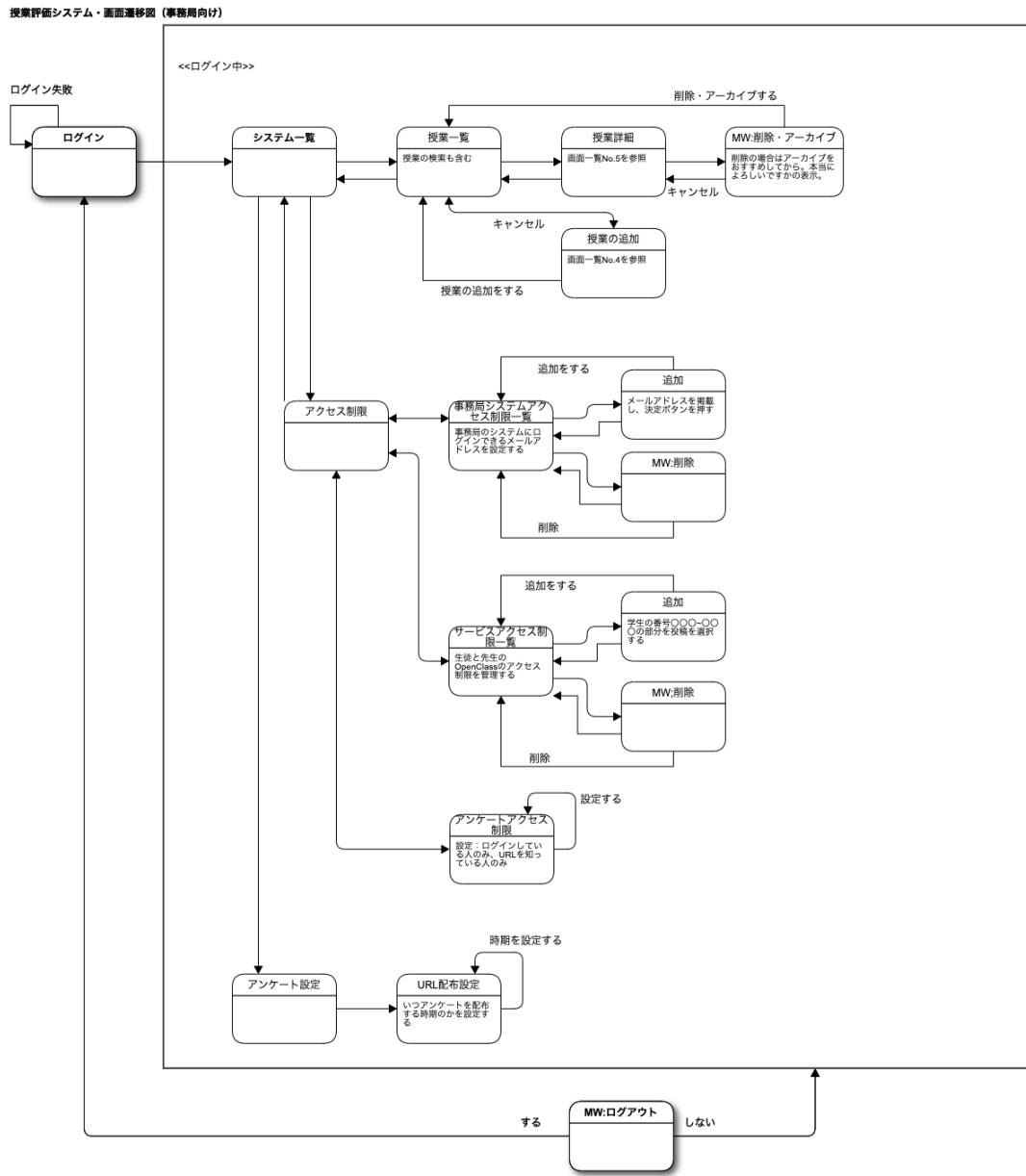
授業評価システム・画面一覧（事務局向け）

No.	画面図	概要	その他の機能	備考
1	ログイン	システムにログインする		
2	ログアウト	システムをログアウトする		
3	授業一覧	授業の検索や一覧を表示する	授業を追加ボタン	
4	授業追加	新規授業の追加		授業名、先生の名前、メールアドレス、顔画像を登録する。カテゴリも選択できると良い。
5	授業詳細	授業の詳細情報を表示する	授業の削除・アーカイブ	上記の変更・更新ができると尚よし。
6	アクセス制限	ログインできる人の管理をする。	追加・削除	事務局システム、生徒用・先生用の評価サイトのログイン制限
7	アンケート	アンケートURL配信の設定		決まった日程に先生にアンケートのQRコードが送られる
8				
9				
10				
11				

上図：当時作成した [組織側] の画面一覧表

必要なコンテンツ選定や、ユースケース図作成、アクティビティ図作成をしたら、大きくどのようなページが必要になってくるのかがわかりました。それを参考にこのようなページが必要だろう、想定はしていたが必要なさそうだといった部分がわかったので実際に [組織側]・[ユーザ側] での画面一覧表を作成しました。

● 画面遷移図作成

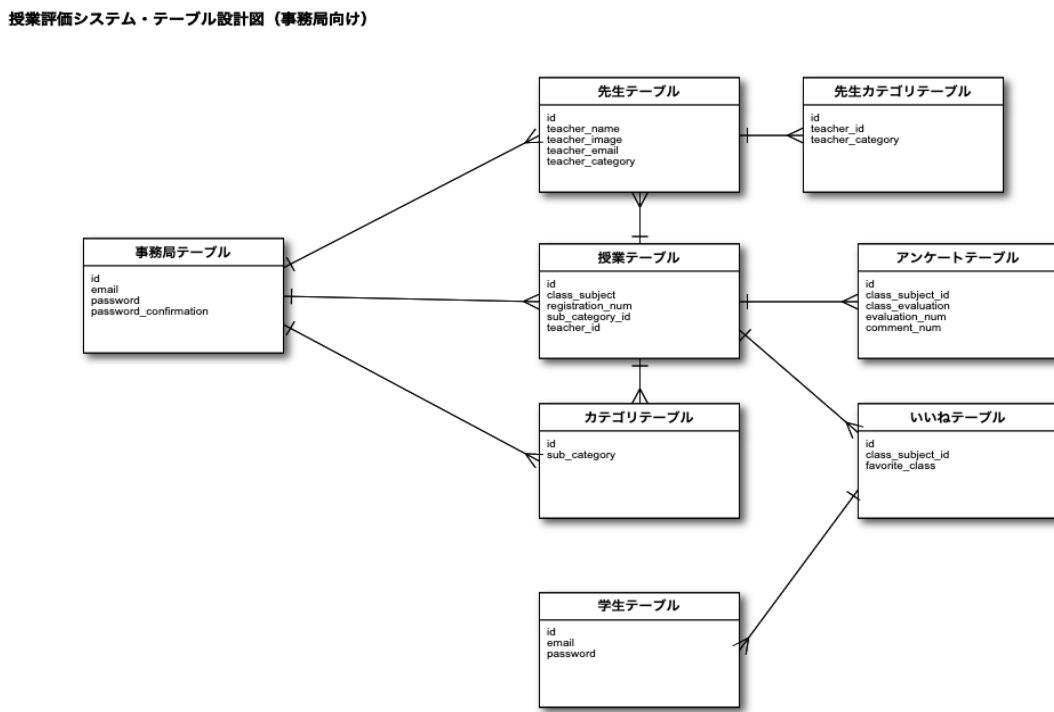


上図：当時作成した [組織側] の画面遷移図

画面一覧表が完成した後は、その画面からどのように遷移するのかを全て図にしました。

これで全体のイメージを綺麗にまとめることができ、どこから開発を進めるべきなのかがわかりやすくなりました。

● テーブル設計図作成



上図：当時作成した [組織側] のテーブル設計図

テーブル設計を行うことで、どのデータとどのデータを紐づけてデータベースを作成すれば良いのかを視覚的にみることが出来ます。

そうすることで、データベースのカラムと呼ばれるものを決定する際に役立ちます。

● 開発技術・環境 選定

作っていくものがより明確化したので、どのような技術や、開発環境が必要なのかがわかつてきました。

● スケジュール

準備段階で最後に行ったのが開発のスケジュールです。

開発期間にどれくらいかかるのかを予想し、予定を考えました。

私たちは 10 月に完成させるというスケジュールで開発を始めました。

結果としては全く完成させることはできなかったのですが、目標を立てないと目処が立たないのでとりあえずの目標を立てるだけでも良いと思います。

● デザイン作成

デザインは全期間を含め1ヶ月はかかったと思います。最初に決めていたデザインから途中で変更することはほとんどできないのでかなり試行錯誤しました。

あまり良くないかもしれません、私たちは組織側から制作をしたので、まず組織側のデザインだけを行い、ユーザ側の開発に入る段階でデザインを開始しました。

デザインを担当した三崎はデザインの授業をとったこともなく、デザインに触れたのはゼミが初めてだったので何度も矢野と相談をしながら進めていました。

● 準備段階まとめ

私たちはこの段階を終わらせるのがゼミの中でも早かったのではと感じています。

三崎が矢野に共同制作の話を持ちかけた時点からかなりモチベーションが高かったために、6月の終わりの時点ではここまで終わらせていました。

早く進めることができが必ずしも良いわけではないかもしれません、元々 "OpenClass." は制作量が多いため、ここまで軽快に進めることができたのはよかったです。

また、準備段階では色々なアイデアを出す部分だと思うので、個人の制作よりもペースが早く、多くの思考を巡らせることができたのではと思っています。

新規のサービスを考えるのは非常に面白かったです。

制作者が行ってきたこと /

■ 開発段階

● 開発分担の明確化

三崎がjQueryやデザインの学習を行なっていたため、フロントエンド・マークアップ(Webアプリケーションで接触ユーザが目に触れる部分)を担当しました。

矢野はPHPやMySQLの学習を行なっていたため、バックエンド(Webサーバー側やデータベースのシステムなど、ユーザの目に見えない部分)を担当しました。

私たちは同じプログラミング言語でも違う分野を学んでいたため、開発前の役割分担はスムーズに行うことができました。

● 開発手法・方向性

私たちは、アジャイル開発のエクストリーム・プログラミング(Extreme Programming- 以下 XP)という手法を用いて開発を進めることにしました。XPとは、事前に立てた計画よりも途中変更などの柔軟性を重視する方法です。

私たちはチームでの開発が初めてであったため、準備段階では想定していなかったことが起こり、その都度修正をしながら、結果的にこのような手法になりました。

● 開発環境

開発環境については、AWSのCloud9の中にPHPのフレームワークのLaravelを環境構築して、進めました。

Cloud9には統合開発環境機能が付いており、アプリケーション開発にとって必要なツールや機能を全て兼ね備えており、ブラウザ上での操作になるため、端末の環境に左右されることなく使用できます。

Cloud9はペアプログラミングの機能も付いているため、データや作業の共有が容易にでき、私たちが懸念していた、お互いのローカル環境のちょっとした相違よって、作業が進まない事を解消してくれた為、効率的に作業を進める事ができました。

● 開発期間中

私たちはローカル環境を共有して開発し、密にコミュニケーションを取って制作を行ってきました。ほぼ毎日のように連絡を取り合いながら試行錯誤していました。

最初の段階では、矢野が一つの機能を追加したら、次に三崎がその機能に対するデザインを加えていくことを交互にやっていました。

機能が増えていけば完成に近づいていく実感をやりがいにして、励んでいましたが、機能が増えればそれに伴うバグが増えるのも開発期間中には付き物でした。

その中で私たちが、大事にしていた価値観は卒業制作を通じてスキルアップをし、成長することでした。私たちは、まだまだプログラミングのスキルが発展途上だったからです。

開発期間の中盤から終盤にかけて、徐々に Web アプリケーションとして形になってきながらは、お互いのどちらかが、修正点に気付いたら課題点を指摘し合いながら修正していくようになっていました。

しかし開発期間の終盤にかけては、開発をすればするほど「まだ完成ではない」と考え込んでしまい、終わりなき道のように感じてしまい、お互いのモチベーションも徐々に下がり開発スピードも低下してしまいました。

そこで私たちは、実際に生徒側にアプリケーションを使ってもらうために、期間を設けて完成を目指すことで開発スピードを回復させました。

実際に生徒にアプリを使ってもらった時には、エンジニアとしての喜びをお互い感じる事ができました。

■ モチベーション

● 結成当初

○ モチベーション：100%

私たちが結成したのは 2020 年の 12 月の後半ごろでした。

三寄から卒業制作における共同制作を矢野に持ちかけました。元々三寄はフロントエンドを学習しており、矢野がバックエンドを学習していたので、2 人が合わさることで大きい規模のサービスを作ることができるのではないかと考えていました。

意気投合してからは色々なサービスにおけるアイデアを出す「アイデア出し」をたくさんしていました。

とにかく人のために何かできることを探していました。

当時は何も考えずただ意見を出すだけだったので非常に面白くこれから始まる制作に心を踊らされる超高モチベーションだったと思います。

体感では一番面白く、モチベーションも一番高い時期だったのではないかと思っています。

● 準備段階

○ モチベーション：90%

具体的に何を作っていくのかが定まってきた段階です。

こちらも 2 人で "OpenClass." というサービスのために何ができるのかを考えるだけだったため、非常に楽しく、モチベーションもかなり高かったです。

cacoo, mind meister, Google ドキュメントというクラウドサービスを使いサービスをより良いものにしていくという部分や、zoom や oVice, workle, Line といったチャット・会話ツールを使いお互いの状況をシェアする、新たなツールを使って作業をしていることが非常に楽しかったです。

ただ、具体的にサービスを考えていると様々な課題が上がってくるのが現実で、正解がわからない中で、答えを探し続けるのはかなり難しかったです。

それでも完成するときの想像を掻き立てながら進めるのはモチベーションが上がりました。

● デザイン段階

○ 初期 モチベーション：95%

色々なサービスやデザインを見て "OpenClass." にどのように活かせるのか・ゼミで学んだ UI/UX を意識し、使う人の気持ちを考えながらの作業が多かったため楽しいパートでした。

実際にデザインを進める中で、あらゆる工夫を組み込んでこのようなギミックがあつたら面白いだろうなということをデザインに組み込んでいく工程も面白くモチベーションにつながっていました。

○ 中期 モチベーション：80%

作業をどんどん進めていく中でどれが正解なのかがわからず、悩む時間が増えてきました。実際にこのデザインやギミックをコーディングに移すにあたり、技術的にクリアできるのかなどの考察も増えてきました。

手が止まる時間が増えてくる中で人になかなか相談できなかったという部分でモチベーションが少し下がってしました。

○ 後期 モチベーション：68%

FB のないデザインという作業が段々と疲れてきました。

中期のように正解がわからないのもそうですが、誰かに相談をするべきだったと反省しています。

さらに、予定していたスケジュールが迫っていると段々と嫌気が差してしまって、完全にモチベーションが下がってしまいました。

新鮮さがなくなってくるというのが良くなかった部分だと思い、途中の段階で一旦切り上げ、開発に移行することを決めました。

● 前半まとめ

私たちは共同作業で進めていたため、相手に相談しながら作業を進めていくことができました。それが返ってゼミの先生や、友人に相談しないという点に繋がり FB をもらえず、ここを直した方がいいのか！という気づきをモチベーションに変えることができなかったのかと思いました。

それでも総合して準備段階ではとても楽しく作業をしていたと思います。

● 開発段階

○ 初期 モチベーション：90%

初期はデザインしたものをコーディングすることや・プログラミングの開発の環境を整えたりと、今までとは違う作業だったのでワクワクしていました。

実際にシステム開発やコーディングをはじめて、一つ一つの機能が完成することがとても嬉しくモチベーションが上がっていました。自分たちの技術が劣らず立ち止まるタイミングがあったものの、初期の段階ではモチベーションに影響することはありませんでした。とにかく作業をするのが楽しくモチベーションは継続していました。

○ 中期 モチベーション：70%

中期では開発という作業が慣れてきた時期もあり、「ただ作る作業」という気持ちになってしましました。

さらに、機能としての矛盾点を消すために新たな機能を追加しているうちに、自分たちが生み出したシステム的なエラーが生じてしまい、作ってはエラー作ってはエラーという部分を繰り返していました。

それでも新たな機能が完成すると嬉しく完成に近づいていると思うとモチベーションが上がりました。

○ 後期 モチベーション：40%

アプリケーションとして操作できるようになってきた時点で満足する雰囲気になっててしまい、ここからさらに制作を続けていくモチベーションが下がってきました。

ゼミの先生方に協力してもらい、授業評価のアンケートを実施させていただいた際に、アンケートの集計ができた時点で完成と勘違いしてしまい、さらにモチベーションが下がってしまいました。

ウェブアプリケーションの特性上、高みを目指せばいくらでも作業ができるため自分たちで終わりを決めなくてはならないため、あやふやな状態が続きました。

最終的には着地点をしっかりと設け、そこを目指してしっかりとモチベーションを立て直しました。

レポート

■ 所感

卒業制作を通し、サービスを生み出すという観点で問題点を一つ一つ解決していくことの重要さを痛感しました。

問題を後伸ばしにすればするほど、大きな問題となり突き当たったり、モチベーションが下がるといった状態になってしまいました。

準備段階で考えていたサービスの内容も問題を後回しにしたことで、解決できなかった問題もいくつかあるのではないかと思います。

ただそれでも、共同制作という形態で制作してこれたのは何よりもプラスになってきたと感じています。

元々私たちそれぞれが習得していた技術が違っていたために、お互いの穴を埋める形で結成できたことはもちろんですが、何よりも励まし合いという点でチームである意味が非常に大きかったなと思います。

一方のモチベーションが落ちている時は一方が励まし、お互いを助け合ってきたと思います。

もちろん個人制作では自由にという点では有意義なものがありますが、私たちは互いができる技術や分野・思考が全く違うからこそ埋め合える点が多くあり、卒業制作では非常に貴重な経験になったと感じます。

私たちは起業思考があります。

かなり簡易的かもしれません、ある意味今回の制作ではそれを体験できたような気がします。

今後、どのようなタイミングで卒業制作で培ったモノを発揮するかはわかりませんが、いずれにせよ、一年かけていい経験をできたと感じています。

■ 考察

現状の "OpenClass." では、実際に使ってもらうようなアプリケーションまでは至らなかつたなと感じています。

一番は「使い勝手」にあると思っています。

実際にスクレイピングという技術を用いて、授業全てのデータをまとめたところ、教員が約 180 名、授業が約 820 コマという結果でした。

それだけ多くのデータを大学の既存のアプリケーションだけではなく、"OpenClass." でも操作しなければならないと考えると、その分かなりの機能性に長けていなければ難しいと感じました。

API の開発等で、大学既存のアプリケーションにアクセスできるような技術があれば、それはまたより良い製品になると感じました。

セキュリティを考えても、個人情報を扱っている時点でかなり難しいものでした。

私たちは、教育機関（今回は大学）から学生に向けてアンケートを取ることで、新規授業の増設に対応でき、信憑性のあるデータも取れると考えていました。

しかし、実際に教育機関がどのように学生にアンケートを提示すれば回答してくれるのかという問題点は一向に解を出すことができませんでした。

現在の DHU 既存の FS では提出が出席の条件であり、その条件がいかに重要か、それほどどの理由がなければ記入は難しいと感じました。

ただ、Web & UI/UX ゼミの教員 2 名の授業でアンケートを行ったときには、普段の FB では得られないような解答を得られたのではないかと感じたこともあり、解答をしてくれるという条件が揃いさえすれば、面白いサービスになるのではないかと私たちは考えています。

同様のサービスを提供している、「楽单らくだ」「楽天みんなのキャンパス」がどのような理由で学生個人主体で授業評価等を記入してもらう形態にしたのかは定かではありませんが、質や規模を考えなければ、サービスを始めるコストが低く抑えられるのかなと感じました。

■ 感想

● 三寄 遥樹

まず一言で「長かった」と私は感じました。

コロナ禍ということもあり、家にこもり作業を続けるということは私にとってはかなり疲弊するものがありました。

ただ、そんな中でも Line や oVice, workle, zoom というツールに非常に助けられたなと感じました。共同制作で一番大事なのは間違いなくコミュニケーションなので、こんな状況下でもコミュニケーションをたやすく密に連絡をとり続けられたのはすごいことなんじゃないかと思ってしまいます。

ここ一年で話さなかった合計日数は 7 日もないくらいでは？と思います。

私は何もかも独学だったという点で当初はどうなることかと思いましたが、長い時間を使える卒業制作でかなり成長できたと思います。

技術もそうですが、自分なりに物事の考え方を整理する力を養うことができたなと思っており、色々な成長を期待できるのが卒業制作なのだと実感しました。

解決できなかった問題点が多く残されてしまったことは残念ですし、自分の追い込みの甘さも原因だと感じています。

特に私の方がモチベーションが下がってしまうことが多かったなと感じていますが、パートナーである矢野がその時に毎回完成の話をしてくれるとすごくワクワクしたのを思い出します。

最初に共同制作を持ちかけた時からそうですが、かなりポジティブ思考の矢野には毎回助けられていました。本当に感謝です。

私個人的には、共同制作をお勧めしたいと思います。

● 矢野 涼

私は卒業制作を通じて、思うようにいかない経験を沢山できた事が一番良かったと思います。

正直卒業制作を始める前は、「共同制作だし、すぐに開発し終わるだろう」と高を括っていました。

実際始めてみると、計画していた通りに行かない事も多々あり、その度に三崎と連絡をとりながら軌道修正するためには、どうしたらいいのかを話し合いました。

卒業制作で一番の思い出は、こういった毎日のコミュニケーションです。

長期にわたって一つの作品に向き合うのは初めての経験でしたし、何より完成に向けてただただ邁進しようとした過程は貴重な経験になりました。

四年生の時の思い出と言えば、「共同制作での卒業制作です。」と 10 年後にも言ってると思います。笑

私もできたら共同制作をお勧めしますが、どういった物を作るかよりもパートナー選びが一番重要だと思います。

私はパートナーが三崎だからここまで来れたと思います。三崎はどちらかというと細かい性格なので、私が抜けている部分をいつも補完してくれました。

あと卒業制作を始める前に比べて、三崎のプログラミングの伸びがエグかったです。笑

最後の方は javascript に関してなんでも書けるのではないかと思いました。笑

こうしてここまで来れたことには感謝です。ありがとう。そしてお疲れ様でした。

■ 謝辞

本制作を進めるにあたり、Web & UI/UX ゼミ教員である小松学史 准教授、栗谷幸助 准教授には多くの適切なご指導を賜り、担当されている授業での授業評価アンケートにご協力いただき誠に感謝いたします。

Web & UI/UX ゼミの学生の皆様にもゼミ内授業評価のご協力をいただきましたこと感謝申し上げます。

また、渡部晋作先生には新たなアイデアを頂き、実装することこそ叶いませんでしたが、卒業制作を始めるにあたり非常に良い刺激を与えてくださいました。ありがとうございました。

■ 参考文献 / ツール

● 参考文献

○ chart.js 実装

- ・<https://misc.0o0o.org/chartjs-doc-ja/>
- ・<https://code-database.com/knowledges/83>

○ Laravel

- ・<https://qiita.com/makies/items/2ab24188e7f8482bfddc>
- ・<https://qiita.com/mindwood/items/baf756b1ac430c7e4e4d>
- ・<https://nana-engineer.com/2021/02/13/laravel-db-ajax/>

○ AWS

- ・<https://qiita.com/kono-hiroki/items/86af406d95f325b32e4e>
- ・https://docs.aws.amazon.com/ja_jp/elasticloadbalancing/latest/application/application-load-balancers.html

● ツール (開発以外)

○ cacoo

- ・<https://cacoo.com/ja/features>

○ mind meister

- ・<https://www.mindmeister.com/pages/ja/mind-mapping>

○ oVice

- ・<https://ovice.in/ja/>