

2016 年度 卒業論文

面接練習支援システムの開発

大阪産業大学 デザイン工学部 情報システム学科
情報教育システム研究室

13H050 高橋幸大

目次

1	はじめに	1
2	目的	2
3	大学新卒者採用における面接評価	3
4	面接練習支援システムの概要	4
4.1	動作	4
4.2	面接練習の動画の準備	4
4.3	振り返り	4
5	本研究で開発したシステムの内容	5
5.1	本システムの概要	5
5.2	システムの仕様	8
5.3	YouTube Data API	9
5.4	YouTube Iframe API	10
6	検証内容	12
6.1	検証の参加者	12
6.2	方法	12
7	結果と考察	14
7.1	結果	14
7.2	考察	15
8	結論	22
8.1	今後の課題	22
9	ソースコード	25
9.1	Controller	25
9.2	Model	39
9.3	View	46
9.4	Mailers	69
9.5	Helper	72
9.6	Assets	75
9.7	Library	85
9.8	Task	88
9.9	Config	89

1 はじめに

面接試験は就職活動の中でも、最も臨機応変に対応することが必要とされている。質問に対し、正確に答えるという高度な要求に対して、臨機応変に回答するための練習が必要である。しかし、学生の就職活動の中で面接練習に対する意欲の低さが見られ、練習することを軽視していることがある。その理由では、面接練習ができる環境とフィードバックが得られる環境がないことの2つと考えられる。面接練習をするためには質問を問い掛ける人と質問に答える人がその場に集まり、面接練習をすることでフィードバックから改善する練習方法がある。しかし、従来の練習方法では練習時に人が集まった時でしか練習ができず、フィードバックはその場でしか得れないという問題がある。

第2章ではこの問題をふまえて本研究の目的について述べる。第3章では大学新卒者採用における面接評価 [1] に基づいての評価について述べる。第4章では本研究で開発したシステムの概要を述べる。第5章では本研究で開発したシステムの内容を述べる。第6章では本研究で検証の内容を述べる。第7章では本研究で開発したシステムの結果、その考察とともに述べる。第8章には研究の成果とともに今後の課題についてまとめる。

2 目的

面接練習が軽視されている理由として、面接練習をするための環境が構築されていないと考えられる。面接練習をするためには練習に取り組むための環境と改善へと繋げるためのフィードバックが必要である。

本研究の目的は、面接練習ができる環境とフィードバックが受け取れる環境を構築するための Web アプリケーションの開発である。本システムによって、学生の面接練習の意識向上を図ることを目指す。

3 大学新卒者採用における面接評価

面接評価には言語的情報、非言語的情報にカテゴライズされた項目 [1] を評価対象とする。言語的情報とは志願者の発言行為によって発信される情報であり、コンテンツとメタ情報にカテゴライズされる。コンテンツとは言語的情報の中での発言の内容であり、主に基礎能力が問われる。基礎能力とは分析結果では、様々な評価事項の中でも必ず含まれているものがあり、そのものを単独コードと定義する。その単独コードの中には知識・技能、価値観と名付けられた。その中でも知識・技術は、志願者に直接その知識について尋ねることや学習した内容や学習量を尋ねることで判断している。本研究では学生同士の面接練習であるため、同じ環境で生活しているため知識・技術についての判断が可能と考えた。また本研究では口頭のみ面接練習のため知識・技術の項目を知識に変更した。メタ情報とはコンテンツ以外の要素すべてを指す。非言語的情報とは言語によって発信されるものを除いた情報である。

本研究では表 1 の項目を評価対象とする。

表 1 カテゴリと項目

カテゴリ	項目
コンテンツ	知識
メタ情報	言葉遣い 質問の答えの正確さ 表現力 対人能力 志望の強さ
非言語的情報	印象 魅力

また各評価項目に対して点数を割り当てる。点数とその点数に対する意味を表 2 に示す。

表 2 評価の点数と意味

点数	意味
5	率先して採用したい
4	採用しても良い
3	特に問題はないが、判断しにくい
2	採用するには小さな問題がある
1	採用するには大きな問題がある

また点数以外の評価としてコメントを残せるようにする。

4 面接練習支援システムの概要

本研究では、学生同士による面接練習の環境を構築と共に練習を促し、改善のサイクルを組むためのものである。そのため、面接練習をしている学生の動画を撮影し、学生同士で評価をシェアできるようにする。ロールプレイング手法に基づいた面接練習支援システムの研究 [2] を参考に Web アプリケーションを開発した。

4.1 動作

以下にシステムの動作を示す。

1. 本システムを使用するためのアカウントを作成する
2. 評価対象となる学生の面接動画を準備する
3. 面接練習の動画をシステムに登録する
4. 登録された動画を視聴し、評価をする
5. 出演している動画の評価を閲覧し、振り返る

4.2 面接練習の動画の準備

面接練習の動画を準備するにあたって、検証に参加する学生に事前に資料を配布した。

以下の順番で動画の準備をした。

1. 面接練習の対象となる企業の情報を収集
2. 自己分析
3. 面接練習の動画を撮影

上記の項目を次項より説明する。

4.2.1 面接練習の対象となる企業の情報を収集

面接練習をするにあたって志望をする企業を 1 社とその企業の特徴などの情報収集を各学生が事前に行う。志望する業界、企業は不問である。

4.2.2 自己分析

面接練習で自分のことを話をする際の自己分析を行う。自己分析をするにあたって家族、友人に話を聞くように指導をした。

4.2.3 面接練習の動画を撮影

面接は 1 対 1 で行い、10 分以内の練習を行った。質問はあらかじめ準備をしていた項目とその返答に対する発展した質問を行った。詳細は第 6 章にて述べる。

4.3 振り返り

面接練習の動画を本システムに登録し、出演者を含み検証に参加している学生からの評価を行う。その評価から次回の面接練習をする際に注意をする点などを確認し、面接の質の向上を計る。

5 本研究で開発したシステムの内容

5.1 本システムの概要

全体の概要図を図 1 に示す。主に 4 つの機能を実装した。次項より各機能の解説を行う。すべての機能にユー

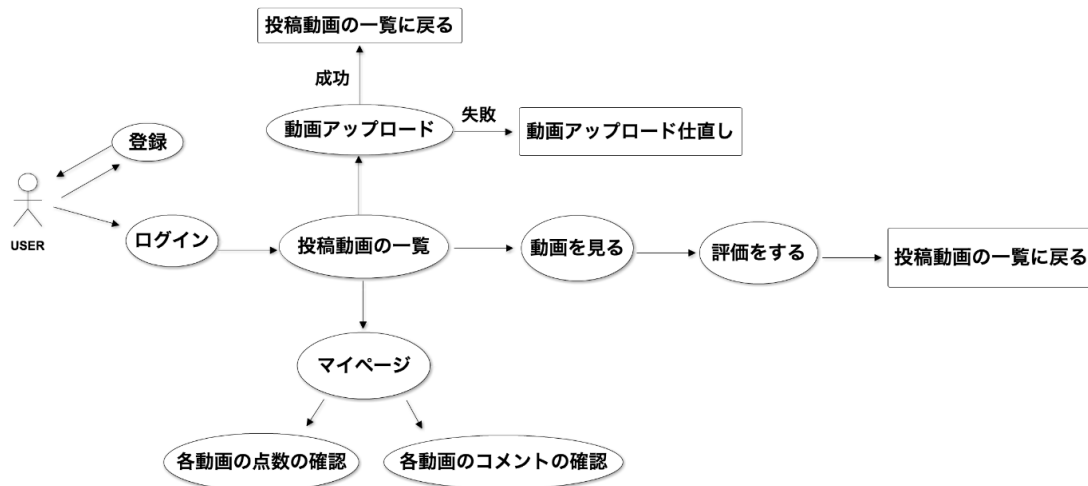


図 1 本システムのユーザ視点でのユーザ登録、動画のアップロード、動画の閲覧、評価の確認の操作

ザ認証が必要である。

1. ユーザ登録・認可・認証
2. 動画登録
3. 評価機能
4. 評価の集計

5.1.1 ユーザ登録・認可・認証

ユーザ登録には名前、メールアドレス、パスワードを必要とする。本学の学生向けの Web アプリケーションのため、ユーザ登録には FQDN^{*1}が@ge.osaka-sandai.ac.jp のメールアドレスのみ登録が可能である。登録する際に本人確認のために認可用のメールを送信する。認可メールには認可を得るための URL が記載されているため、その URL 先に移動することで認可される。本システムの機能はすべてユーザ認証が必要である。面接練習の情報を外部に漏れることを防ぐためである。ユーザ認証にはユーザ名とパスワードが必要である。ユーザ名は本学の先頭文字「s」+ 学籍番号としている。パスワードは各人自由に設定ができるようにしてあり、認証時には Bcrypt というハッシュ関数を用いている。Bcrypt によって得られたハッシュ値と、データベースに登録されているユーザ情報のハッシュ値とで照合し、ユーザ認証を行う。ユーザ登録の例を図 2、認可メールの例を図 3、ユーザ認証の例を図 4 にそれぞれ示す。

^{*1} Fully Qualified Domain Name

ユーザアカウントを作成

名前

フルネームをお願いします

メールアドレス

s学生番号@ge.osaka-sandai.ac.jpのメールアドレスを入力してください。

パスワード

*6文字以上にしてください

ユーザ登録

Copyright©2016 Kodai Takahashi

図 2 ユーザ登録の例 名前 (フルネーム)、メールアドレス (ge.osaka-sandai.ac.jp)、パスワード (6 文字以上) を入力する。不正なデータが入力された際、もう一度入力を求める。



図 3 認可メールの例 ユーザ名と認可を得るの URL が記載されている。認可を得るためには URL 先に接続することで本システムが使用可能になる。

5.1.2 動画登録

動画を登録する際、アップロード画面を設けた。アップロードをする際に必要な項目としてタイトル、評価対象者、動画ファイルである。タイトルには面接練習の対象となる企業名、評価対象者は面接練習をしている学生、動画ファイルは Content-type にて定義されている video/mp4 と video/quicktime のファイルを入力する。動画のアップロードには YouTube Data API を使用し動画がアップロードをし、その結果をデータベースに登録する。動画登録の例を図 5 に示す。

5.1.3 評価機能

第 3 章に述べた評価項目に従って評価をする。評価は一度だけで可能であり、取り消しは不可能であるため誤送信を避けるために確認画面を設けてある。

評価機能と同時に評価をする際に面接動画の視聴機能を搭載しており、面接動画を視聴しながら評価が可能である。面接動画の視聴には YouTube Iframe Player API を使用している。図 6 に動画視聴機能、図 7 に評価機

ログイン

ユーザ名
s00h000

パスワード
.....

ログイン

[アカウントを作成](#)

Copyright©2016 Kodai Takahashi

図 4 ログインの例 ユーザ名、パスワードを入力する。ユーザ名とパスワードを照合し、一致していた場合ログインが完了する。不正なデータが入力された際、もう一度入力を求める。

アップロード

タイトル
産犬

評価対象者
高橋幸大

動画
ファイルを選択 0.mp4
movファイルもしくはmp4ファイルで10分以内の動画を指定してください

アップロード

Copyright©2016 Kodai Takahashi

図 5 動画登録の例 タイトル、出演者、動画ファイルを入力する。正当なデータの場合動画が登録される。不正なデータが入力された際、もう一度入力を求める。

能の例をそれぞれ示す。

5.1.4 評価の集計

評価機能によって投稿された評価を動画毎に集計、平均を抽出し、出演者が各評価項目の点数を棒グラフで確認ができる。点数は棒グラフで表示されており、x 軸は平均点、y 軸は動画のタイトルである。コメントは上記に面接動画が表示されており、下部にはコメントが表示されてる。出演している動画のタイトル、投稿日時などの情報、また各動画で得た全体評価の点数の平均値が表示されている例を図 8 に示す。各動画で得た言葉遣い、質問の答えの正確さの点数の平均値を表示している例を図 9 に示す。各動画で得た対人能力、知識の点数の平均値を表示している例を図 10 に示す。各動画で得た表現力、印象の点数の平均値を表示している例を図 11 に示す。各動画で得た魅力、志望の強さの点数の平均値を表示している例を図 12 に示す。出演している動画、面接練習の動画に対してのコメントを表示している例を図 13 に示す。

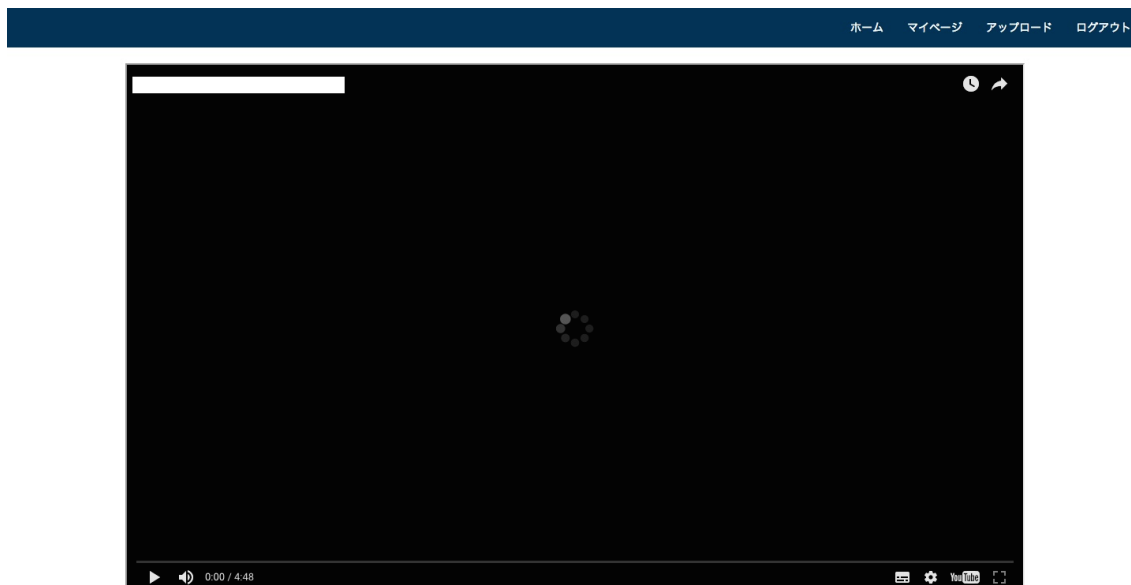


図 6 評価画面 上部 面接練習の動画を視聴するページ YouTube Iframe Player API によって YouTube プレイヤーを使用している

総合評価	<input type="text" value="3"/>
言葉遣い	<input type="text" value="3"/>
質問の答えの正確さ	<input type="text" value="3"/>
対人能力	<input type="text" value="3"/>
知識	<input type="text" value="3"/>
表現力	<input type="text" value="3"/>
印象	<input type="text" value="3"/>
魅力	<input type="text" value="3"/>
志望の強さ	<input type="text" value="3"/>
面接に対するコメント	<input type="text"/>

※必須ではありません。255文字以内のコメントでよろしくお願ひします。

図 7 評価画面 下部 上部に表示されている動画を視聴し、各評価の項目に適当な値をシステムに送る

5.2 システムの仕様

システムはプログラミング言語 Ruby で作成し、Ruby on Rails^{*2}、Unicorn^{*3}、Nginx^{*4}、PostgreSQL^{*5}、YouTube Data API、YouTube Iframe API で利用しシステムを構築した。

^{*2} <http://rubyonrails.org/>

^{*3} <http://unicorn.bogomips.org/>

^{*4} <http://nginx.org/>

^{*5} <http://www.postgresql.org/>

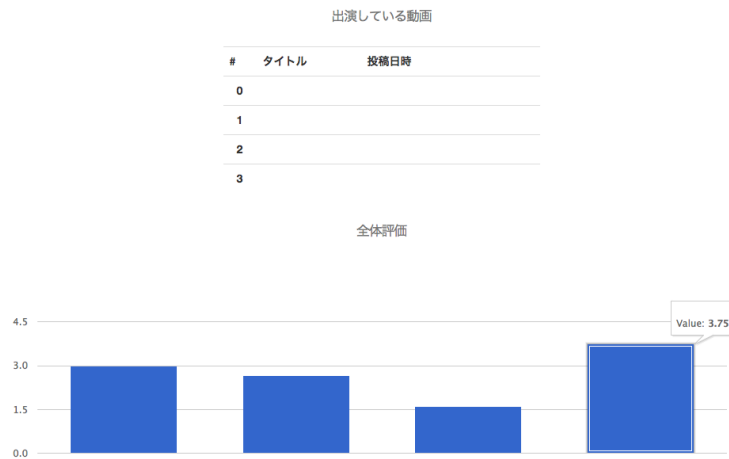


図 8 スコア確認の画面 1 上部には出演している動画のタイトル、投稿日時が表示されている。下部は各動画で評価された全体評価の平均値による棒グラフが表示される。

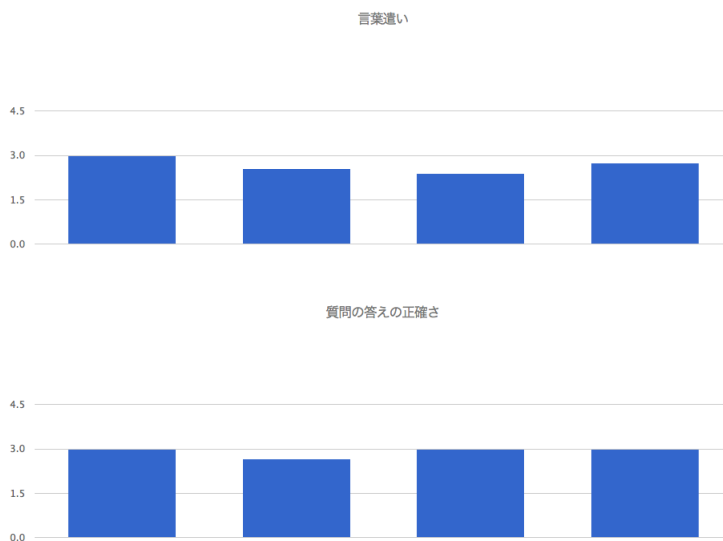


図 9 スコア確認の画面 2 上部は各動画で評価された言葉遣いの平均値による棒グラフが表示している。下部は各動画で評価された質問の答えの正確さの平均値による棒グラフが表示している。

5.3 YouTube Data API

YouTube Data API [3] は Google が提供している API アプリケーションに YouTube の様々な機能を扱える API である。動画のアップロードユーザの再生リストなどを管理することができる。開発者が目的どおりにサービスを使用できることと、不当にサービス品質を低下させたり他のリソースへのアクセスを制限したりしてしまうアプリケーションの作成を防ぐことを目的としてクォータを使用する。その為、以下の制限がある。

1. 動画の長さは 10 分以内
2. 動画のサイズは 1 GB 以下

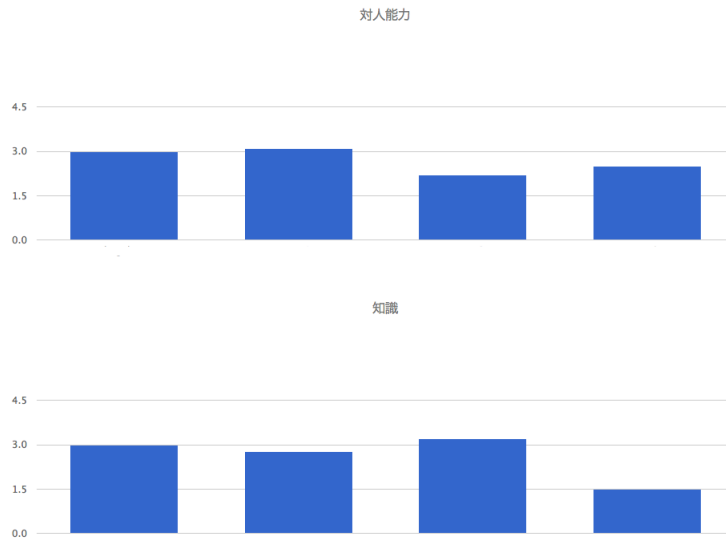


図 10 スコア確認の画面 3 上部は各動画で評価された言葉遣いの平均値による棒グラフが表示している。下部は各動画で評価された言葉遣いの平均値による棒グラフが表示している。

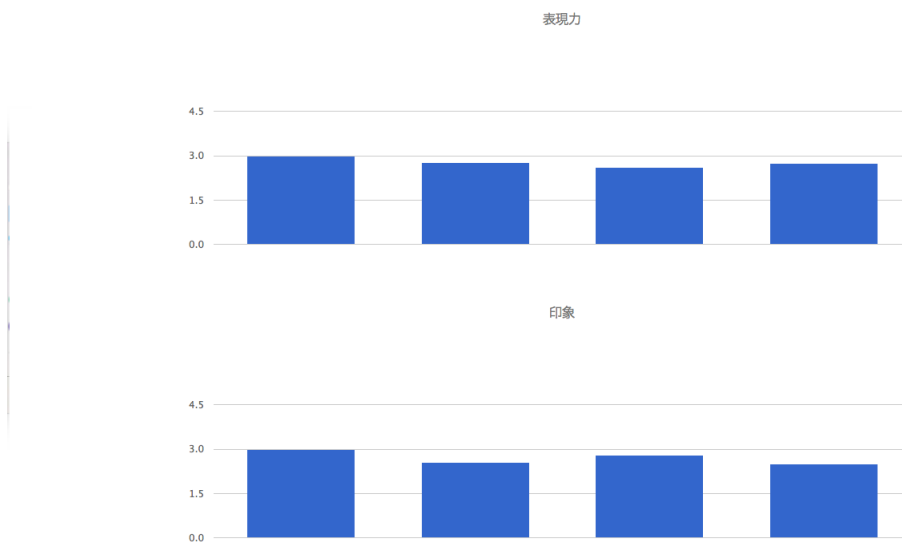


図 11 スコア確認の画面 4 上部は各動画で評価された表現力の平均値による棒グラフが表示している。下部は各動画で評価された印象の平均値による棒グラフが表示している。

3. 1つのアカウントで保持できる動画の最大数は 2000

本研究では事前に撮影された動画をアップロードをすることや、Youtube Iframe API に使うための動画 ID を取得するために使用する。

5.4 YouTube Iframe API

YouTube Iframe Player API [4] は YouTube 動画プレーヤーをウェブサイトに埋め込み、JavaScript でプレーヤーを制御ができる。ウェブページに Flash オブジェクトを埋め込む Flash や JavaScript のプレーヤー API とは異なり、YouTube Iframe Player API はコンテンツをページの Iframe タグに投稿する。本研究では、本システ

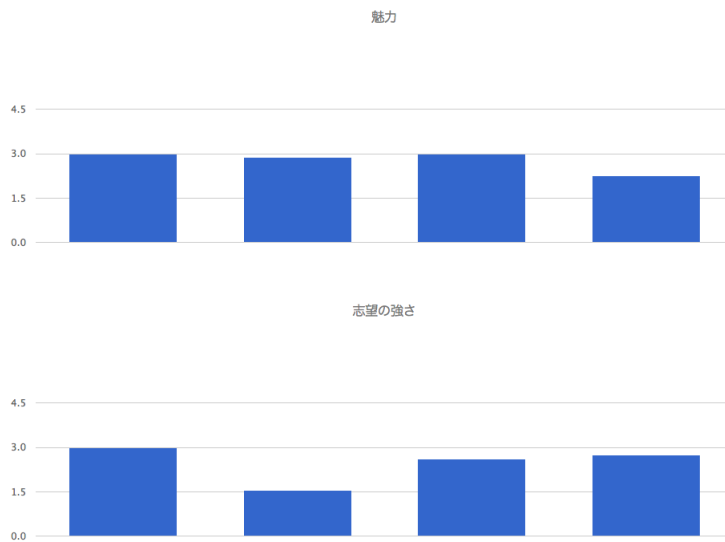


図 12 スコア確認の画面 5 上部は各動画で評価された印象の平均値による棒グラフが表示している。下部は各動画で評価された志望の強さの平均値による棒グラフが表示している。

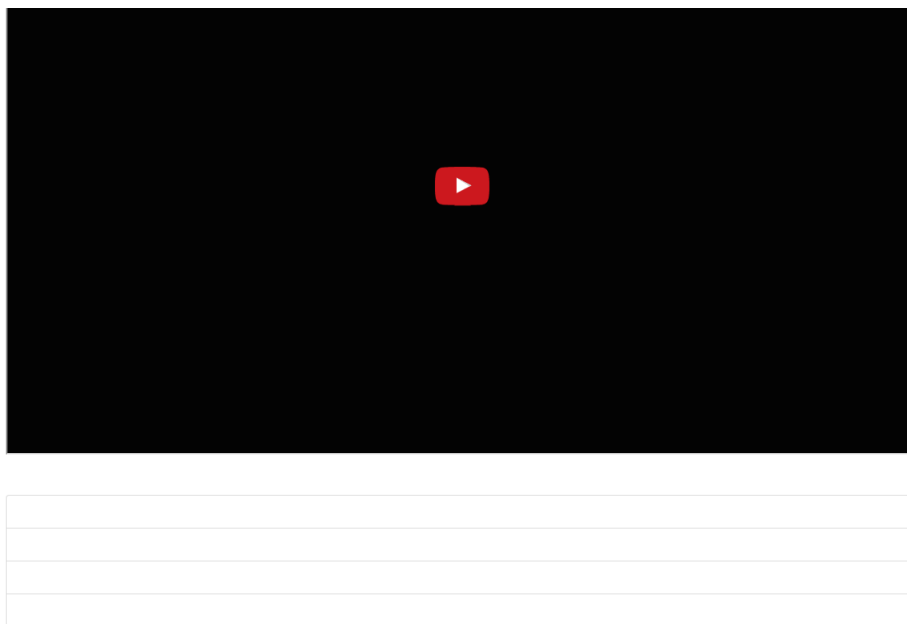


図 13 コメント確認の画面 上部にはコメントが残されている面接練習の動画が表示している。下部は上部の面接練習に対するコメントが表示している。

ムで YouTube Data API によってアップロードされた動画を再生するために YouTube Iframe API を使用する。

6 検証内容

6.1 検証の参加者

1. デザイン工学部情報システム学科 3 年生・・・2 人
2. デザイン工学部情報システム学科 2 年生・・・3 人

検証の参加者は計 5 人である。また面接動画に対する評価のみの参加者は上記の 5 人を除き 10 人である。

6.2 方法

6.2.1 概要

検証の参加者 5 人には面接練習を学生同士で 2 回行い、面接の動画に対して評価を付けてもらうことにした。

6.2.2 質問

面接練習を行う際に必ずしてもらう質問を 4 つ準備をし、第 4.2 で調査した企業を対象に 10 分以内の面接をしてもらった。質問は下記ものである。

1. 自己 PR をお願いします。
2. 学生時代に頑張ったことはなんですか。
3. なぜ弊社を希望されましたか。
4. 今後この業界になにが必要だと思いますか。

上記の質問をしてもらい気になる点などを再度質問してもらうことにした。

6.2.3 面接練習の環境

1. 1800 × 900 の長方形テーブル・・・1 つ
2. イス・・・2 席
3. コンパクトデジタルカメラ・・・1 つ
4. コンパクトデジタルカメラのための三脚・・・1 つ
5. ノート PC・・・1 台
6. 鉛筆・・・1 本
7. ボールペンシル・・・1 本
8. A4 サイズの用紙・・・2 枚

質問をする学生と面接を受ける学生は向き合う形でイスに座る。机には動画撮影用に三脚を付けたデジタルコンパクトカメラを面接を受ける学生側に向け、撮影を始める。また質問をする側には A4 サイズの用紙と鉛筆、ボールペンシルを配布し、質問の回答をメモをしてもらうことにした。面接練習の全体を図 14 に示す。また実際の面接練習の様子を図 15 に示す。



図 14 面接練習の全体像



図 15 面接練習の様子 1

7 結果と考察

7.1 結果

今回結果をまとめるにあたって、面接練習支援システムの結果としてデザイン工学部情報システム学科3年生と2年生から各代表一人のデータとアンケートの結果を用いる。

7.1.1 各学生の結果

7.1.2 3年生

出演している動画のタイトル、投稿日時などの情報、また各動画で得た全体評価の点数の平均値が表示されている例を図16に示す。各動画で得た言葉遣い、質問の答えの正確さの点数の平均値を表示している例を図17に示す。各動画で得た対人能力、知識の点数の平均値を表示している例を図18に示す。各動画で得た表現力、印象の点数の平均値を表示している例を図19に示す。各動画で得た魅力、志望の強さの点数の平均値を表示している例を図20に示す。出演している動画、面接練習の動画に対してのコメントを表示している例を図21に示す

3年生の結果から2回目の面接練習の表現力の評価は1回目の面接練習の結果から改善をすることで評価を上げることができている。

7.1.3 2年生

出演している動画のタイトル、投稿日時などの情報、また各動画で得た全体評価の点数の平均値が表示されている例を図22に示す。各動画で得た言葉遣い、質問の答えの正確さの点数の平均値を表示している例を図23に示す。各動画で得た対人能力、知識の点数の平均値を表示している例を図24に示す。各動画で得た表現力、印象の点数の平均値を表示している例を図25に示す。各動画で得た魅力、志望の強さの点数の平均値を表示している例を図26に示す。出演している動画、面接練習の動画に対してのコメントを表示している例を図27に示す。

2年生の結果から2回目の面接練習の言葉遣いの評価は1回目の面接練習の結果から改善をすることで評価を上げることができている。

7.1.4 アンケートの結果

またシステムを検証の参加者にアンケートを行った。アンケートは以下の条件で行った。

- 対象は検証の参加者5名。
- Google社がサービス提供をしているGoogle Forms^{*6}を使ってアンケートを実施。
- 主なアンケート項目は以下の通り
 - － 自分の面接練習のフィードバックが受けられる環境はどのように感じましたか？
 - － 他人の面接練習を評価をする環境はどのように感じましたか？
 - － これからも利用したいと思いますか？

このアンケートで得た回答をもとに結果をまとめる。

問1の自分の面接練習のフィードバックが受けられる環境はどのように感じましたか？という項目には良かったもしくは良くなかったの2択の回答を設けた。結果は以下の表3に示す。回答の理由をまとめると以下の結果が得られた。

- いただいたフィードバックを次の面接に活かすことができる

^{*6} <https://www.google.com/intl/ja-jp/forms/about/>

- 練習を行うごとにダメだった点などの差がわかるので非常にタメになった

問3の他人の面接練習を評価する環境はどのように感じましたか？という項目には良かったもしくは良くなかったの2択の回答を設けた。結果は以下の表4に示す。回答の理由をまとめると以下の結果が得られた。

- 自分と違う意見、考え方があり勉強になった
- 面接未経験でもどういった点を気をつけて動画を見れば良いのかがわかりやすい

問5のこれからも利用したいと思いますか？という項目には利用したいもしくはしたくないの2択の回答を設けた。結果は以下の表5に示す。

表3 問1の結果

項目	回答人数
良かった	5人
良くなかった	0人

表4 問3の結果

項目	回答人数
良かった	5人
良くなかった	0人

表5 問5の結果

項目	回答人数
利用したい	5人
利用したくない	0人

以上がアンケートの結果である。

二つの結果から本研究である面接ができる環境とフィードバックが受け取れる環境の構築はできた。



図16 3回生 スコア確認の画面1

7.2 考察

面接練習ができる環境とフィードバックが受け取れる環境の構築はできたが、以下の改善が必要と考える。詳細は今後の課題に示す。

- 面接練習に対するモチベーションの維持
- 面接練習に対するアプローチの仕方
- 面接動画の長さの制限
- 面接動画の評価を企業ごとにカテゴリ分け

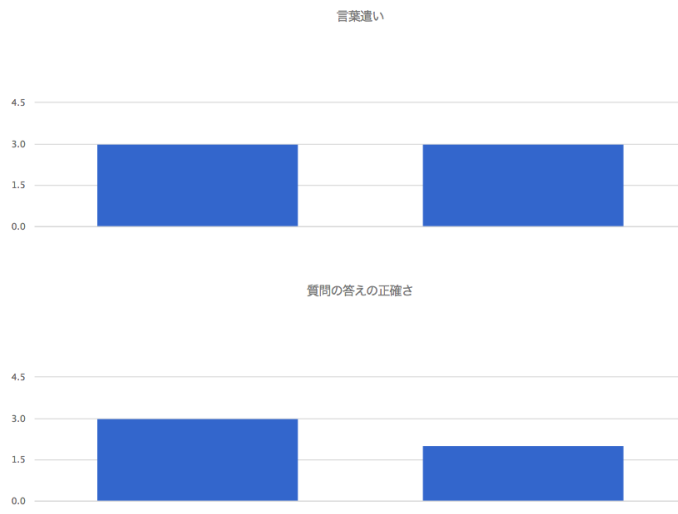


図 17 3 回生 スコア確認の画面 2



図 18 3 回生 スコア確認の画面 3

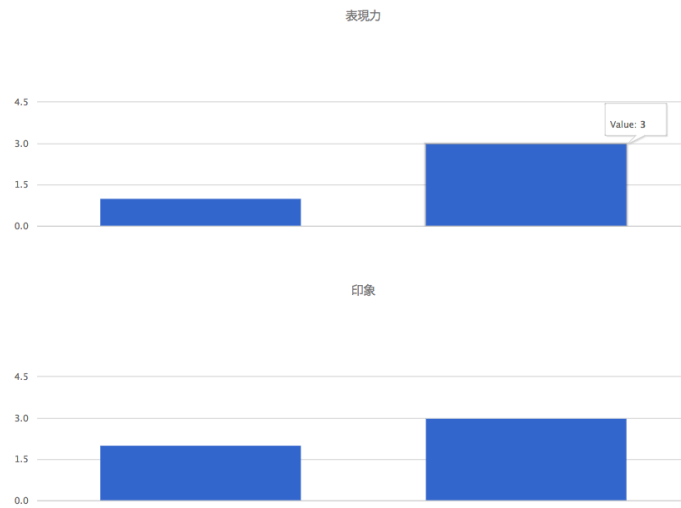


図 19 3 回生 スコア確認の画面 4

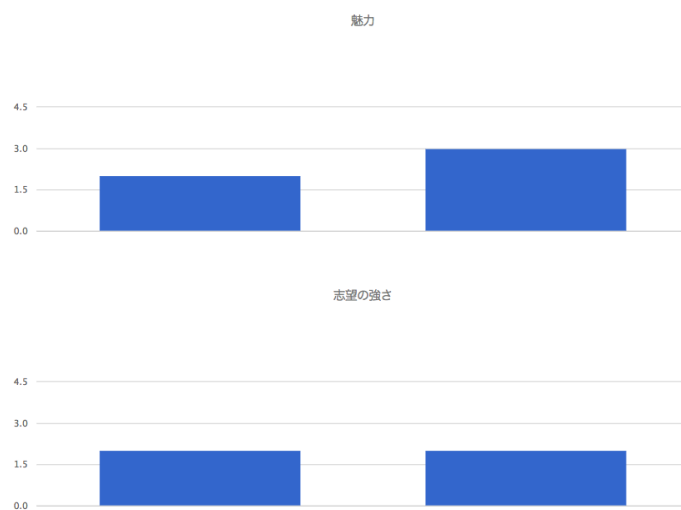
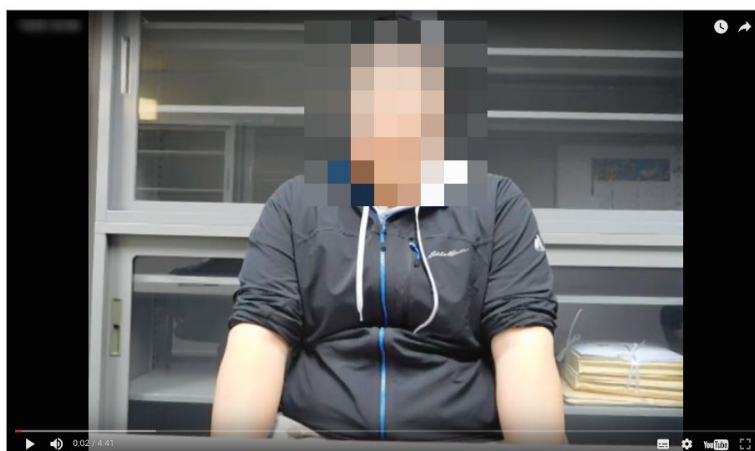


図 20 3 回生 スコア確認の画面 5



初めにしてはよく話せてと思います。まず話すときに気にすることはやってきたことでも自分がどのように動いたのか、など 3回生の名前) という人を伝えられるように話してみてください。それだけで一気に面接がよくなりますよ、それと一番大事なことは元気を出して言ってみよう!!」とコメントがされている。

Copyright©2016 Kodai Takahashi

図 21 3 回生 コメント確認画面 「初めにしてはよく話せてと思います。まず話す時には気にすることはやってきたことでも自分がどのように動いたのか、など (3 回生の名前) という人を伝えられるように話してみてください。それだけで一気に面接がよくなりますよ、それと一番大事なことは元気を出して言ってみよう!!」とコメントがされている。

出演している動画

#	タイトル	投稿日時
0		
1		

全体評価

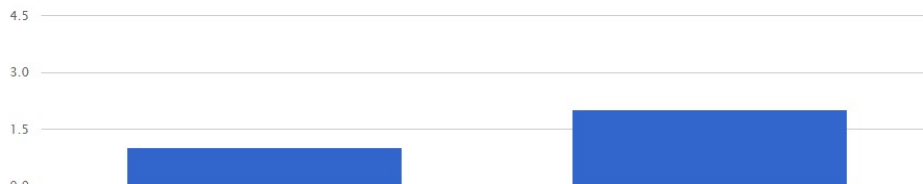


図 22 2 回生 スコア確認の画面 1

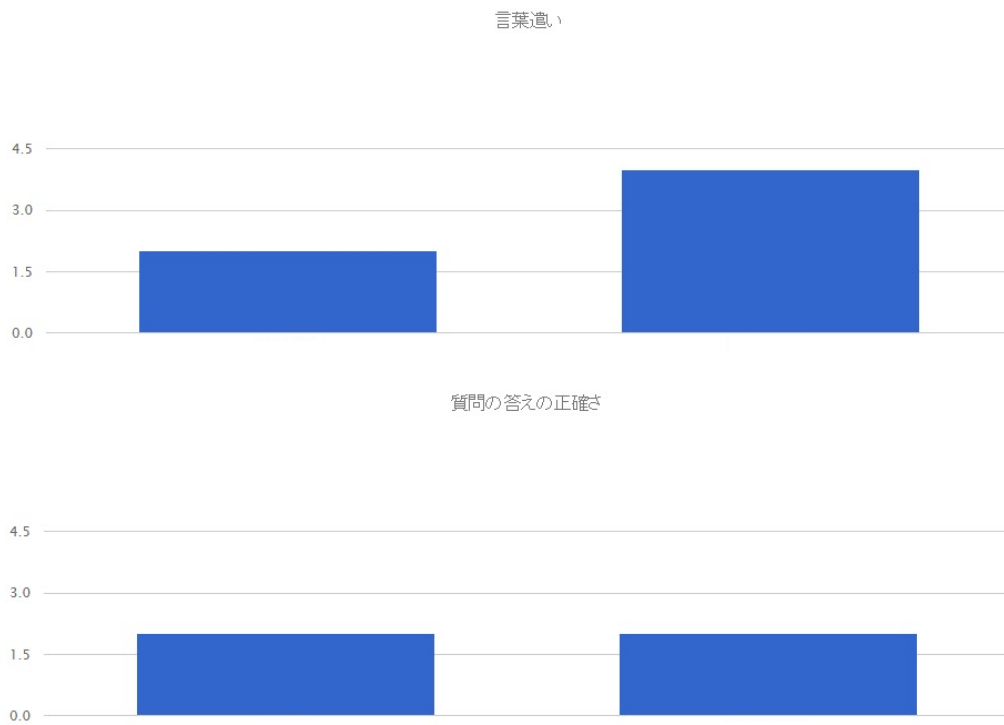


図 23 2 回生 スコア確認の画面 2

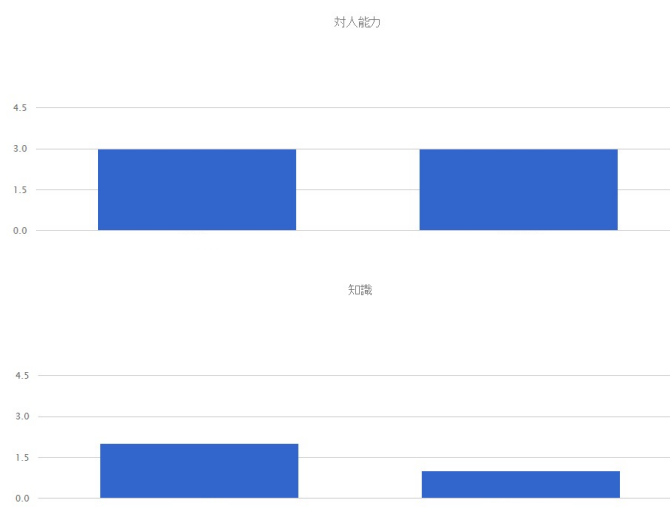


図 24 2 回生 スコア確認の画面 3

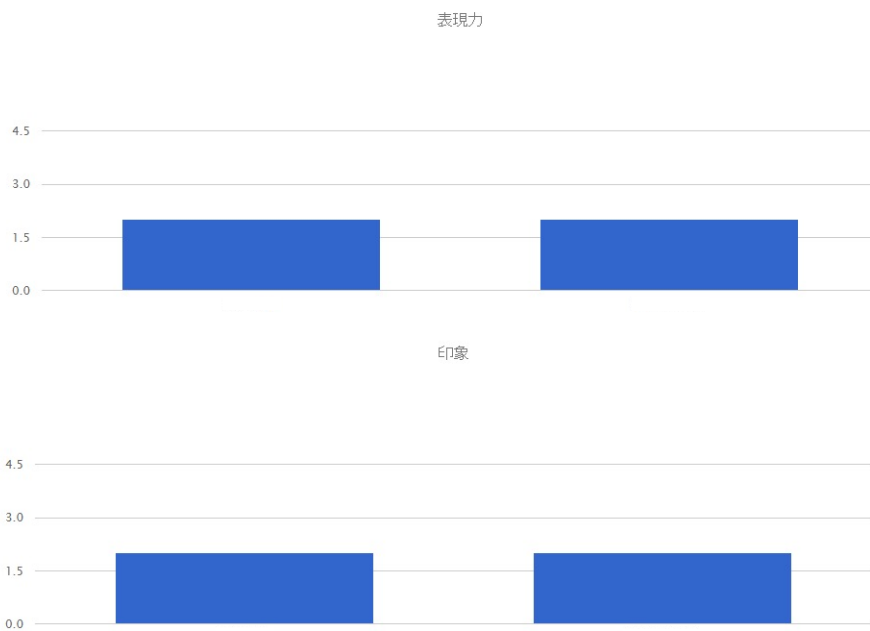


図 25 2 回生 スコア確認の画面 4

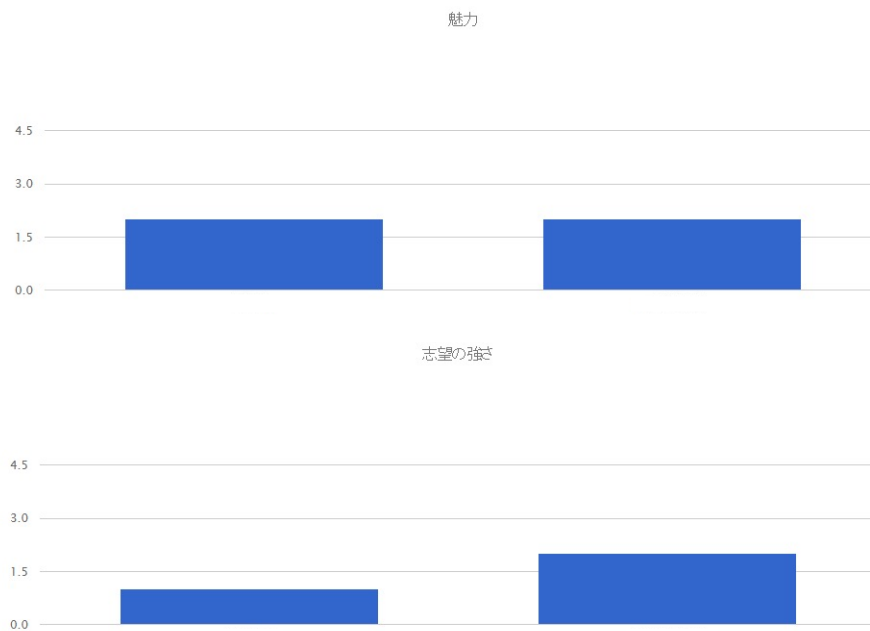
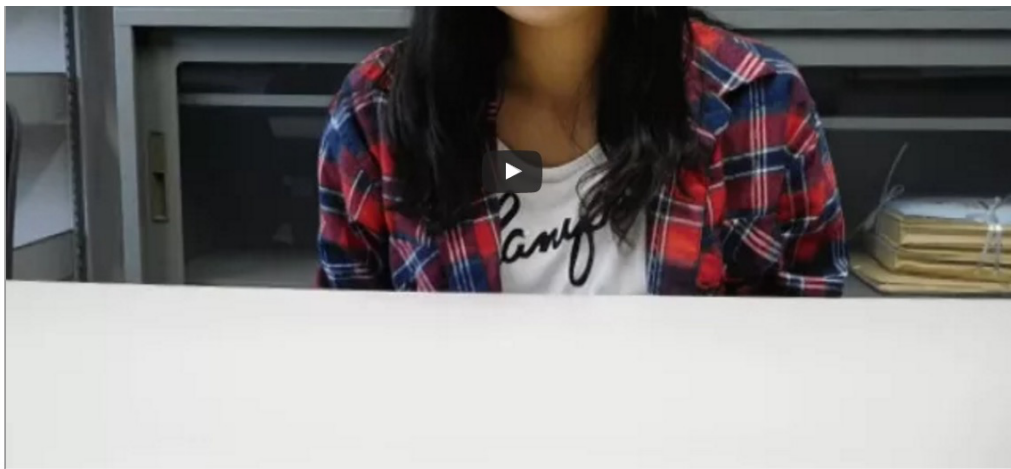


図 26 2 回生 スコア確認の画面 5



大学名と名前の答え方は「〇〇大学〇〇学部〇〇学科から来ました〇〇です。本日はよろしくお願ひします。」的なことが言えたらOK!!なぜ弊社が良いのかというところでは、よかったところをもっと具体的に言ってみよう!目線が下に行くことが多いので質問を投げってくる人の目と合わせよう!

図 27 2 回生 コメント確認の画面「大学名と名前の答え方は「 大学 学部 学科から来ました
です。本日はよろしくお願ひします。」的なことが言えたら OK!! なぜ弊社が良いのかというところでは、よ
かったところをもっと具体的に言ってみよう! 目線が下に行くことが多いので質問を投げってくる人の目と合わ
せよう!」とコメントされている。

8 結論

本研究では面接練習ができる環境とフィードバックが得られる環境を構築するための Web アプリケーションを開発を行った。その結果、フィードバックを受けることとフィードバックをすることで自分を振り返り、どのような点を気をつけるべきかを見つけることが可能となった。また先行研究 [2] との違いを以下の示す。

- 面接練習の動画が随時ネットワーク上で視聴ができる。
- 面接練習の評価が棒グラフにて可視化されている。
- 動画の視聴、動画のアップロード、動画の評価、評価の確認が本システムのみで完結する。

上記の違う点の一つである面接練習の動画が随時ネットワーク上で視聴ができることでは、先行研究 [2] は Ustream^{*7} という動画配信サービスを使用している。動画をローカル環境に動画をダウンロードしての視聴は可能ではあるが、ネットワーク上で視聴する際、Ustream に動画をアップロードするには有料会員になる必要があり手間がかかる。これにより検証の参加者が自分の面接練習、もしくは他者の面接連取を視聴する際に動画ファイルを所持しなければならない。本システムでは YouTube^{*8} を使用することで無料会員のまま動画をアップロードができ、動画はネットワーク上で視聴ができるため、動画ファイルを所持する必要がない。

8.1 今後の課題

今後の課題は、実装ができなかった要素の実装及び、面接練習の評価の有効活用、学生個人の能力に依存せず面接練習を活発に行えるようにすることである。上記の内容から以下の課題が考えられる。

8.1.1 モチベーションの維持

面接練習をする際、人前で話す行為に苦手意識が強く感じられた。これには学生同士の信頼度にも依存すると思われるが、システム側では面接練習により得られた評価だけではなく、別の要素を加えることが必要である。加える要素としてはゲーム要素であるステータスとレベル、ポイント、ルールなどを組み込むことでより活発に行えると考えられる。

8.1.2 面接練習をするためのアプローチ

Web アプリケーションにより面接練習の環境の構築及び、フィードバックが受け取れる環境は構築できたが、面接練習をするためのアプローチを改良していくことが望ましい。

8.1.3 面接動画の時間の制限

本稿では面接動画に対して評価をするということではできたが、面接動画に時間の制限がある。本番の面接では短いものから長いものまでであるため、様々な面接練習ができるようにすることが望ましい。

8.1.4 面接動画を企業ごとにカテゴリ分け

本研究では面接練習をするための企業を 1 つだけ選出してもらい使用してもらった。面接練習をする企業は 1 つだけとは限られないため、企業ごとに面接練習の評価が必要である。

*7 <http://www.ustream.tv/>

*8 <https://www.youtube.com/>

謝辞

本研究を進めていく上で、担当教員である大垣斉准教授に御指導及び御協力を戴きました。また、情報教育システム研究室の所属の学生及び卒業生の方々、team.andrew ML に参加されている皆様方に御助言を賜りました。深く感謝の意を表します。

そして、実験参加者である情報教育システム研究室所属の学生、及び大阪産業大学デザイン工学部情報システム学科の学生の皆様に感謝の意を表します。

参考文献

- [1] 岩崎千裕. 大学新卒者採用における面接評価の構造. 労働制作研究, 2009.
- [2] 井上直大 高橋大樹 中村亮太 上林憲行. ロールプレイング手法に基づいた面接練習支援サービスの研究 SNS と改善サイクルによる面接力向上の効果検証. 情報処理学会全国大会講演論文集, 2012.
- [3] Youtube data api — google developers. <https://developers.google.com/youtube/v3/>.
- [4] iframe 組み込みの youtube player api リファレンス — youtube iframe player api — google developers. https://developers.google.com/youtube/iframe_api_reference?hl=ja.

9 ソースコード

9.1 Controller

Listing 1 コントローラーの共通処理 (application_controller.rb)

```
class ApplicationController < ActionController::Base

  class Forbidden < ActionController::ActionControllerError; end
  class IpAddressRejected < ActionController::ActionControllerError; end

  # Prevent CSRF attacks by raising an exception.
  # For APIs, you may want to use :null_session instead.
  protect_from_forgery with: :exception
  include SessionsHelper
  include VideosHelper

  if !Rails.env.development?
    rescue_from Exception, with: :render_500
    rescue_from ActiveRecord::RecordNotFound, with: :render_404
    rescue_from ActionController::RoutingError, with: :render_404
  end

  def routing_error
    raise ActionController::RoutingError.new(params[:path])
  end

  def render_404(e = nil)
    logger.info "Rendering_404_with_exception:_#{e.message}" if e

    if request.xhr?
      render json: { error: '404_error' }, status: 404
    else
      format = params[:format] == :json ? :json : :html
      render template: 'errors/error_404',
             formats: format,
             status: 404,
             layout: 'application',
             content_type: 'text/html'
    end
  end
end
```

```
def render_500(e = nil)
  logger.info "Rendering 500 with exception: #{e.message}" if e
  Airbrake.notify(e) if e

  if request.xhr?
    render json: { error: '500_error' }, status: 500
  else
    format = params[:format] == :json ? :json : :html
    render template: 'errors/error_500',
      formats: format,
      status: 500,
      layout: 'application',
      content_type: 'text/html'
  end
end
end
end
```

Listing 2 認証の処理 (auths_controller.rb)

```
class AuthsController < ApplicationController

  def create
    auth = request.env['omniauth.auth']
    p auth
    user = YoutubeUser.find_or_initialize_by(uid: auth['uid'])
    user.token = auth['credentials']['token']
    user.refresh_token = auth['credentials']['refresh_token']
    user.name = auth['info']['name']
    user.save
    session[:youtube_user_id] = user.id
    flash[:success] = "Welcome, #{user.name}!"
    redirect_to '/videos/upload'
  end

  def failure
    render text: "Sorry, but the following error has occurred: #{params[:
      message]}.
    Please try again or contact administrator."
  end
end
```

Listing 3 セッションの処理 (sessions_controller.rb)

```
class SessionsController < ApplicationController

  def new
    if logged_in?
      redirect_to videos_path
    else
      @title = 'ログイン'
      @login_form = {}
    end
  end

  def create
    user = User.find_by(user_id: params[:id])
    if user && user.authenticate(params[:password]) && user.activated
      log_in user
      remember user
      redirect_to videos_path
      #ログイン成功時
    else
      #ログイン失敗時
      flash.now[:danger] = 'ユーザ名もしくはパスワードが正しくありません'
      @login_form = {id: params[:id]}
      render 'new'
    end
  end

  def destroy
    log_out
    redirect_to login_path
  end

end
```

Listing 4 ユーザーの処理 (users_controller.rb)

```

class UsersController < ApplicationController

  before_action :log_in?, only: [:index]
  protect_from_forgery :except => [:create]

  def score
    user_id = current_user.id
    video_ids = Video.video_id_list
    @videos = Video.video_title_list(user_id, video_ids)
    @overall_evaluation_data = Review.overall_evaluation_ave(user_id,
      video_ids)
    @speech_data = Review.speech_ave(user_id, video_ids)
    @exact_answer_data = Review.exact_answer_ave(user_id, video_ids)
    @communication_data = Review.communication_ave(user_id, video_ids)
    @knowledge_data = Review.knowledge_ave(user_id, video_ids)
    @expression_data = Review.expression_ave(user_id, video_ids)
    @impression_data = Review.impression_ave(user_id, video_ids)
    @attractive_data = Review.attractive_ave(user_id, video_ids)
    @wish_data = Review.wish_ave(user_id, video_ids)
  end

  def video_list
    @videos = Video.where(target_user_id: current_user.id).order('id_DESC').
      page(params[:page]).per(4)
  end

  def show
    @video_id = params[:id]
    @comments = Review.get_comment(current_user.id, @video_id)
  end

  def new
    @registration_form = {}
  end

  def create
    user = User.new(
      email: params[:email],
      name: params[:name],

```

```

    password_digest: params[:password])
  if user.valid?
    user.save
    ActivationMailer.activation_user(user.email).deliver_later
    render text: '入力されたメールアドレスにを送信したため、にクリックした後ログインが可能と
      なります。URLURL'
  else
    if user.errors[:email].include?('is_invalid')
      flash.now[:'bg-danger'] = 'メールアドレスは学生番号@ge.osaka-sandai.ac.のみ登
        録ができますjp'
    elsif user.errors[:email].include?('has_already_been_taken')
      flash.now[:'bg-danger'] = 'すでに登録済みです'
    elsif user.errors[:password_digest].include?('is_too_short_(minimum_is_6
      _characters)')
      flash.now[:'bg-danger'] = 'パスワードは文字以上です6'
    else
      flash.now[:'bg-danger'] = '問題が起きました。もう一度やりなおしてください。'
    end
    @registration_form = params
    render 'new'
  end
end

def activation
  activation_digest = params[:id]
  if user = User.find_by(activation_digest: activation_digest)
    user.update(activated: true, activat_at: Time.now)
    redirect_to login_path, flash: {success: '本登録が完了しました。ログインしてくだ
      さい。'}
  else
    render text: '不正なパラメータです。'
  end
end

private

def log_in?
  redirect_to '/login' unless logged_in?
end

```

end

Listing 5 動画の処理 (videos_controller.rb)

```

class VideosController < ApplicationController

  before_action :log_in?
  protect_from_forgery :except => [:upload]

  def index
    @videos = Video.order('id_DESC').page(params[:page]).per(4)
  end

  def show
    begin
      Video.find_by!(video_id: params[:id])
      @title = 'player'
      @video_id = params[:id]
      @titles =
        %w全体評価( 言葉遣い応答の的確さ対人能力知識・技術表現力印象人間的魅力対人印象好感
          )
      @reviewd = Review.find_by(reviewed_user: current_user.id, video_id:
        @video_id)
    rescue => e
      render text: e
    end
  end

  def new
    @upload_form= {}
  end

  def upload
    if check_params(params)
      upload_datas = {
        auth_token: current_youtube_user.token,
        video_size: params[:file].size,
        video_title: params[:title],
        video_content_type: params[:file].content_type
      }
      upload = VideoUpload.new(upload_datas)
    end
  end
end

```

```

if upload.setup!
  if upload.start!(@video)
    store_uploaded_data(params[:title], upload.video_id, params[:user])
    notification_uploaded_all_user(
      user_id: params[:user],
      video_title: params[:title],
      video_id: upload.video_id
    )
    redirect_to videos_path
  else
    flash.now[:'bg-danger'] = '問題が起こったため、もう一度アップロードしてください'
    render 'new'
  end
else
  token = refresh_token!
  current_youtube_user.update(token: token)
  flash.now[:'bg-danger'] =
    'の有効時間が過ぎているため、再発行しました。tokenももう一度アップロードしなおしてくだ
      さい。'
  render 'new'
end
else
  @upload_form = { title: params[:title], user: params[:user] }
  render 'new'
end
end

private

def log_in?
  redirect_to '/login' unless logged_in?
end

def check_params(params)
  if params[:file].nil?
    flash.now[:'bg-danger'] =
      '動画ファイルがありません。もう一度入力しなおしてください。'
    return false
  elsif User.find_by(id: params[:user]).nil?
    flash.now[:'bg-danger'] =

```

```

        '評価対象者の選択に謝りがあります。もう一度選択しなおしてください。'
    return false
end
@video = params[:file].read
content_type = ContentType.content_type(@video)
unless content_type.include?('quicktime') || content_type.include?('mp4')
    flash.now[:'bg-danger'] =
        'もしくはのみアップロードができます。もう度確認してください。MOVMP41'
    return false
end
true
end

def refresh_token!
    client = Google::APIClient.new
    client.authorization = Signet::OAuth2::Client.new(
        client_id: ENV['YOUTUBE_CLIENT_ID'],
        client_secret: ENV['YOUTUBE_SECRETS'],
        refresh_token: current_youtube_user.refresh_token,
        token_credential_uri: 'https://accounts.google.com/o/oauth2/token'
    )
    client.authorization.refresh!
    client.authorization.access_token
end

def store_uploaded_data(title, video_id, target_user_id)
    upload_user_id = current_user.id
    begin
        Video.create(
            title: title,
            video_id: video_id,
            upload_user_id: upload_user_id,
            target_user_id: target_user_id
        )
    true
rescue
    false
end
end
end

```

```
def notification_uploaded_all_user(user_id:, video_title:, video_id:)
  send_user = User.all_user_email
  send_user.each do |user|
    VideoMailer.notification_uploaded(
      user_id: user_id,
      video_title: video_title,
      video_id: video_id,
      email: user.email
    ).deliver_later
  end
end
end
```

Listing 6 評価の処理 (reviews_controller.rb)

```

class ReviewsController < ApplicationController

  def create
    begin
      raise ActiveRecord::RecordNotFound if Video.find_by!(video_id: params[:
        video_id]).nil?
      @review = Review.new(
        user_id: Video.find_by(video_id: params[:video_id]).target_user_id,
        reviewed_user: current_user.id,
        video_id: params[:video_id],
        overall_evaluation: params[:overall_evaluation],
        speech: params[:speech],
        exact_answer: params[:exact_answer],
        communication: params[:communication],
        knowledge: params[:knowledge],
        expression: params[:expression],
        impression: params[:impression],
        attractive: params[:attractive],
        wish: params[:wish],
        comment: params[:comment]
      )
      if @review.valid?
        @review.save
        render nothing: true, status: 200
      else
        if params[:comment].size > 255
          respond_to do |format|
            format.json { render json:{ error: "コメントは文字までです。255\n確認し
              た後、もう一度評価してください。" },
              status: 422 }
          end
        else
          respond_to do |format|
            format.json {
              render json:{ error: "点数はからまでの数字です。15\n確認した後、もう一
                度評価してください。" },
              status: 422 }
            end
          end
        end
      end
    end
  end
end

```

```
end
rescue ActiveRecord::RecordNotFound => e
  respond_to do |format|
    format.json { render json: { error: '存在しない動画が指定されていますID' },
      status: 422 }
    end
  rescue => e
    respond_to do |format|
      format.json { render json: { error: '存在しない動画が指定されています
        ID' }, status: 422 }
      end
    end
  end
end
```

Listing 7 静的ページの処理 (static_pages_controller.rb)

```
class StaticPagesController < ApplicationController

  def home
  end

  def help
  end

end
```

9.2 Model

Listing 8 ユーザモデルの処理 (user.rb)

```
class User < ActiveRecord::Base
  attr_accessor :remember_token

  has_many :video, foreign_key: 'upload_user_id'

  # scope :list, lambda{
  #   select('id user_id, name')
  # }

  has_secure_password

  # @ge.osaka-sandai.ac.のみを受け取るための定数jp
  VALID_EMAIL_REGEX = /\A[s|a][0-9]{2}[a-z][0-9]{3}@[ge.osaka-sandai.ac.jp|a4w
    .ise.osaka-sandai.ac.jp]/
  VALID_USER_ID_REGEX = /\A[s|a][0-9]{2}[a-z][0-9]{3}/

  # user
  before_validation { self.user_id = email.sub(/@ge.osaka-sandai.ac.jp|@a4w.
    ise.osaka-sandai.ac.jp/, '') }
  validates :user_id, presence: true, length: { maximum: 7 },
  format: { with: VALID_USER_ID_REGEX}

  # #はダウンケースにするemail
  before_save { self.email = email.downcase }
  # #は最大文字に制限name30
  validates :name, presence: true, length: { maximum: 30 }
  # #は最大文字でemail255@geのみに制限...
  validates :email, presence: true, length: { maximum: 255 },
  format: {with: VALID_EMAIL_REGEX}, uniqueness: true
  # #は文字以上に制限password6
  before_create { self.password_digest = User.digest(password_digest) }
  validates :password_digest, presence: true, length: {minimum: 6 }

  # activation_digest
  before_create {self.activation_digest = User.new_token}
```

```

scope :all_user_email , -> {
  where(activated: true).select (:id , :email)
}

# 受け取った文字列をハッシュ値に返す
def self.digest(string)
  cost = ActiveSupport::SecurePassword.min_cost ? BCrypt::Engine::MIN_COST :
    BCrypt::Engine.cost
  BCrypt::Password.create(string, cost: cost)
end

# ランダムなトークンを返す
def self.new_token
  SecureRandom.urlsafe_base64
end

# 永続的なセッションで使用するユーザをデータベースに記憶する
def remember
  self.remember_token = User.new_token
  update_attribute(:remember_digest, User.digest(remember_token))
end

# 渡されたトークンがダイジェストと一致したらを返す true
def authenticated?(remember_token)
  BCrypt::Password.new(remember_digest).is_password?(remember_token)
end

def forget
  update_attribute(:remember_digest, nil)
end
end

```

Listing 9 動画モデルの処理 (video.rb)

```

class Video < ActiveRecord::Base

  belongs_to :user, foreign_key: 'upload_user_id'

  has_many :reviews, primary_key: 'video_id', foreign_key: 'video_id'

  scope :video_id_list, -> { reorder(:created_at).select(:video_id) }
  scope :video_title_list, -> (user_id, video_ids) {
    where(target_user_id: user_id, video_id: video_ids).select(:title, :
      video_id, :created_at)
  }

  # title
  # 空はなし最大文字数文字 30
  validates :title, presence: true, length: { maximum: 30 }

  # video_id
  # 空はなし
  validates :video_id, presence: true

  # upload_user_id
  #空なし 整数のみ
  validates :upload_user_id, presence: true, numericality: { only_integer:
    true }

  #target_user_id
  #空なし 整数のみ
  validates :target_user_id, presence: true, numericality: { only_integer:
    true }
end

```

```

class Review < ActiveRecord::Base

  belongs_to :video, primary_key: 'video_id', foreign_key: 'video_id'

  scope :reviewed?, -> (user_id, video_id) {
    where(video_id: video_id, reviewed_user: user_id)
  }
  validates :user_id, presence: true

  validates :reviewed_user, presence: true

  validates :video_id, presence: true

  validates :overall_evaluation, presence: true,
  numericality: { only_integer: true }, length: { is: 1 }, format: { with: /\A
    [1-5]/}

  validates :speech, presence: true,
  numericality: { only_integer: true }, length: { is: 1 }, format: { with: /\A
    [1-5]/}

  validates :exact_answer, presence: true,
  numericality: { only_integer: true }, length: { is: 1 }, format: { with: /\A
    [1-5]/}

  validates :communication, presence: true,
  numericality: { only_integer: true }, length: { is: 1 }, format: { with: /\A
    [1-5]/}

  validates :knowledge, presence: true,
  numericality: { only_integer: true }, length: { is: 1 }, format: { with: /\A
    [1-5]/}

  validates :expression, presence: true,
  numericality: { only_integer: true }, length: { is: 1 }, format: { with: /\A
    [1-5]/}

  validates :impression, presence: true,
  numericality: { only_integer: true }, length: { is: 1 }, format: { with: /\A

```

```

    [1-5]/}

validates :attractive, presence: true,
numericality: { only_integer: true }, length: { is: 1 }, format: { with: /\A
    [1-5]/}

validates :wish, presence: true,
numericality: { only_integer: true }, length: { is: 1 }, format: { with: /\A
    [1-5]/}

validates :comment, length: { minimum: 0, maximum: 255 }

scope :overall_evaluation_ave, -> (user_id, video_ids) {
  joins(:video).where(user_id: user_id, video_id: video_ids).group(:title).
    average(:overall_evaluation)
}
scope :speech_ave, -> (user_id, video_ids) {
  joins(:video).where(user_id: user_id, video_id: video_ids).group(:title).
    average(:speech)
}
scope :exact_answer_ave, -> (user_id, video_ids) {
  joins(:video).where(user_id: user_id, video_id: video_ids).group(:title).
    average(:exact_answer)
}
scope :communication_ave, -> (user_id, video_ids) {
  joins(:video).where(user_id: user_id, video_id: video_ids).group(:title).
    average(:communication)
}
scope :knowledge_ave, -> (user_id, video_ids) {
  joins(:video).where(user_id: user_id, video_id: video_ids).group(:title).
    average(:knowledge)
}
scope :expression_ave, -> (user_id, video_ids) {
  joins(:video).where(user_id: user_id, video_id: video_ids).group(:title).
    average(:expression)
}
scope :impression_ave, -> (user_id, video_ids) {
  joins(:video).where(user_id: user_id, video_id: video_ids).group(:title).
    average(:impression)
}

```

```
scope :attractive_ave, -> (user_id, video_ids) {
  joins (:video).where(user_id: user_id, video_id: video_ids).group(:title).
    average(:attractive)
}
scope :wish_ave, -> (user_id, video_ids) {
  joins (:video).where(user_id: user_id, video_id: video_ids).group(:title).
    average(:wish)
}

scope :get_comment, -> (user_id, video_id){
  joins (:video).where(user_id: user_id, video_id: video_id).select(:comment)
}
end
```

Listing 11 YouTube モデルの処理 (youtube_user.rb)

```
class YoutubeUser < ActiveRecord::Base
end
```

9.3 View

9.3.1 Layout

Listing 12 Web ページの共通のページ (application.html.slim)

```
doctype html
html lang="ja"
  head
    meta charset="utf-8"
    meta http-equiv="X-UA-Compatible" content="IE=Edge,chrome=1"
    meta name="viewport" content="width=device-width, initial-scale=1.0"
    title= content_for?(:title) ? yield(:title) : "Interview"
    = csrf_meta_tags
    = stylesheet_link_tag "application", :media => "all"
    = favicon_link_tag 'apple-touch-icon-144x144-precomposed.png', :rel => '
      apple-touch-icon-precomposed', :type => 'image/png', :sizes => '144
      x144'
    = favicon_link_tag 'apple-touch-icon-114x114-precomposed.png', :rel => '
      apple-touch-icon-precomposed', :type => 'image/png', :sizes => '114
      x114'
    = favicon_link_tag 'apple-touch-icon-72x72-precomposed.png', :rel => '
      apple-touch-icon-precomposed', :type => 'image/png', :sizes => '72x72'
    = favicon_link_tag 'apple-touch-icon-precomposed.png', :rel => 'apple-
      touch-icon-precomposed', :type => 'image/png'
    = favicon_link_tag 'favicon.ico', :rel => 'shortcut icon'
    = csrf_meta_tags
    = javascript_include_tag "application"
    = javascript_include_tag "//www.google.com/jsapi", "chartkick"
    /! Le HTML5 shim, for IE6-8 support of HTML elements
    /[if lt IE 9]
      = javascript_include_tag "//cdnjs.cloudflare.com/ajax/libs/html5shiv
        /3.7.2/html5shiv.min.js"

  body
    = render 'layouts/header'
    .container
      = yield
    = render 'layouts/footer'
```

Listing 13 Web ページの共通のヘッダー (.header.html.slim)

```

nav.navbar.navbar-default
  .navbar-header
    button.navbar-toggle data-toggle="collapse" data-target="#top-nav" type=
      "button"
    span.icon-bar
    span.icon-bar
    span.icon-bar
  #top-nav.collapse.navbar-collapse
    ul.nav.navbar-nav.navbar-right
      - if logged_in?
        li = link_to "ホーム", videos_path
        / li = link_to "マイページ", user_mypage_path
        li.dropdown role="presentation"
          a.dropdown-toggle data-toggle="dropdown" href="#" role="button"
            aria-haspopup="true" aria-expanded="false" マイペー
            ジ
          ul.dropdown-menu
            li = link_to "スコアを見る", user_score_path
            li = link_to "コメントを見る", user_video_path
            li = link_to "アップロード", new_upload_path
            li = link_to "ログアウト", logout_path, method: "delete"
      - else
        li = link_to トップページ'', root_path
        li = link_to "ログイン", login_path, method: "get"

```

Listing 14 Web ページの共通のフッター (_footer.html.slim)

```
footer . footer
  small . text-right Copyright&copy;2016 Kodai Takahashi
```

9.3.2 Auths

Listing 15 作成用のページ (create.html.slim)

```
= debug @auth
= debug @result.data
```

9.3.3 Sessions

Listing 16 新規作成用ページ (new.html.slim)

```
.page_header
  h1 ログイン
.row
  .col-md-6.col-md-offset-3
    - flash.each do |key, value|
      p.bg-danger.text-center #{value}
    = form_tag login_path, method: 'post', class: 'form-horizontal' do
      = label_tag 'user-name', 'ユーザ名'
      = text_field_tag :id, (params[:id] || @login_form[:id]), class: 'form-control', required: 'true'

      = label_tag 'password', 'パスワード'
      = password_field_tag :password, '', class: 'form-control', required: 'true'
    .row
      .col-sm-12
        .text-center
          = submit_tag 'ログイン', class: "btn btn-lg btn-primary center-block"
      = link_to "アカウントを作成", user_registration_path
```

9.3.4 Users

Listing 17 閲覧用のページ (show.html.slim)

```
.movie-wrap
  iframe src="https://www.youtube.com/embed/#{@video_id}"

.comment.res_center
- if @comments.empty?
  blockquote
    p コメントはまだありません。
- else
  ul.list-group
    - @comments.each do |comment|
      - unless comment.comment.empty?
        li.list-group-item #{comment.comment}
```

Listing 18 新規作成用のページ (new.html.slim)

```

.page-header
  h1 ユーザアカウントを作成
- flash.each do |key, value|
  p.bg-danger.text-center #{value}
= form_tag create_user_path, method: 'POST', class: 'form-horizontal' do
  .row
    .col-md-6.col-md-offset-3
      = label_tag 'name', '名前'
      = text_field_tag :name, (params[:name] || @registration_form[:name]),
        required: 'true', class: "form-control input-search-word",
        placeholder: 'フルネーム'
      ,
    p
      |
      small フルネームをお願いします

      = label_tag 'mail', 'メールアドレス'
      = email_field_tag :email, (params[:email] || @registration_form[:email]),
        required: 'true', class: "form-control input-search-word",
        placeholder: '学生番号
        s@ge.osaka-sandai.ac.jp'
    p
      |
      small 学生番号s@ge.osaka-sandai.ac.のメールアドレスを入力してください。jp
      = label_tag 'password', 'パスワード'

      = password_field_tag :password, '', required: 'true', class: "form-
        control input-search-word"
    p
      |
      small 文字以上にしてください6
  .row
    .col-xs-4.col-xs-offset-5.mt
      = submit_tag 'ユーザ登録', id: 'create', class: "btn btn-lg btn-primary"
      = image_tag 'loader.gif', id: 'loader', class: 'hidden'

javascript:
$(?#create').click(function() {
  if(confirm("上記の情報でアカウントを作成していいですか?")){

```

```
    if ($('#password').val() != '')
      if ($('#name').val() != '')
        if ($('#email').val().match(/(ge.osaka-sandai.ac.jp|a4w.ise.osaka-
          sandai.ac.jp)/ ))
          $('#loader').removeClass('hidden');
        }else
          return false;
      });
```

Listing 19 スコア閲覧用のページ (score.html.slim)

```

h2 出演している動画
.row
  .video-detail
    table.table.m0
      tr
        th #
        th タイトル
        th 投稿日時
      - @videos.each_with_index do | video, i |
        tr.clickable-row data-href="/videos/#{video.video_id}"
          th.text-center #{i}
          th = link_to "#{video.title}", href: "/videos/#{video.video_id}"
          th #{video.created_at.strftime('%Y/%m/%d %H:%M')}

h2 全体評価
= column_chart @overall_evaluation_data , min: 0, max: 5
h2 言葉遣い
= column_chart @speech_data , min: 0, max: 5
h2 質問の答えの正確さ
= column_chart @exact_answer_data , min: 0, max: 5
h2 対人能力
= column_chart @communication_data , min: 0, max: 5
h2 知識
= column_chart @knowledge_data , min: 0, max: 5
h2 表現力
= column_chart @expression_data , min: 0, max: 5
h2 印象
= column_chart @impression_data , min: 0, max: 5
h2 魅力
= column_chart @attractive_data , min: 0, max: 5
h2 志望の強さ
= column_chart @wish_data , min: 0, max: 5

```

Listing 20 出演している動画リストのページ (video_list.html.slim)

```
.page-header
  h1 出演している面接動画
.row
= page_entries_info @videos
- @videos.each do |video|
  .list-group
    a.list-group-item.active href="/user/videos/#{video.video_id}"
      h4.list-group-item-heading.text-center = video.title
      h4.list-group-item-text.text-right 投稿日時
        #{video.created_at.strftime('%Y/%m/%d %H:%M')}
= paginate @videos
```

9.3.5 Videos

Listing 21 動画リストのページ (index.html.slim)

```
.page-header
  h1 面接動画
.row
= page_entries_info @videos
- @videos.each do |video|
  .list-group
    a.list-group-item.active href="/videos/#{video.video_id}"
      h4.list-group-item-heading.text-center = video.title
      h4.list-group-item-text.text-center 評価対象者
        #{User.find(video.target_user_id).name}
      - unless reviewed?(current_user.id, video.video_id).empty?
        h4.list-group-item-text.text-center 評価済み
      h4.list-group-item-text.text-right 投稿日時
        #{video.created_at.strftime('%Y/%m/%d %H:%M')}
= paginate @videos
```

Listing 22 動画の視聴と評価のページ (show.html.slim)

```

.movie-wrap
  iframe src="https://www.youtube.com/embed/#{@video_id}"
/.form
#result.res_center
- if @reviewd.nil?
  = render 'form', item: @reviewd
- else
  .row
    .center
      p レビューが完了しています。

javascript:
$( '#review_form' )
.on("ajax:success", function(e, data, status, xhr){
  $( '#review_form' ).hide();
  return $( '#result' ).append( '<p>レビューを受け付けました。<a href= "/videos"> ホーム</a> </p>' );
})
.on("ajax:error", function(e, xhr, status, error) {
  var error = xhr.responseJSON.error;
  alert(error);
});

```

Listing 23 動画記録のページ (show.html.slim)

```

.page-header
  h1 アップロード
.row
  .col-md-6.col-md-offset-3
    - flash.each do |key, value|
      p.bg-danger.text-center #{value}

  .form-group
    = form_tag videos_upload_path, multipart: 'true', id: 'video_upload',
      method: 'POST' do
      = label_tag 'title', 'タイトル'
      = text_field_tag :title, (params[:title] || @upload_form[:title]),
        required: true, class: "form-control input-search-word"
      = label_tag 'target', '評価対象者'
      = select_tag :user, options_for_select(get_user_list.map { |user| [
        user.name, user.id ]}, selected: (params[:user] || @upload_form[:
        user])), class: 'form-control'
      = label_tag 'movie', '動画'
      = file_field_tag 'file', accept: "video/quicktime,video/mp4,video/avi"
    p
      |
      small ファイルもしくはファイルで分以内の動画を指定してくださいmovmp410
  .col-xs-4.col-xs-offset-4.mt
    = submit_tag 'アップロード', id: "upload", class: "btn btn-lg btn-primary"
    = image_tag 'loader.gif', class: 'hidden', alt: 'Uploading...', id: '
      loader'
    - unless current_youtube_user
      = link_to 'authorize', '/auth/google_oauth2'

javascript:
  $('#upload').click(function() {
    if(confirm("アップロードをしていいですか?\n動画ファイルが大きいほど時間がかかります。")){
      if ($('#title').val() != '')
        if ($('#file').val() != '')
          $('#loader').removeClass('hidden');
        }else
          return false;
    }
  });

```

Listing 24 動画評価の部分テンプレート (_form.html.slim)

```

= form_tag create_review_path, method: 'post', remote: true, id: 'review_form
', class: 'form-horizontal' do
.form-group.mt
  = label_tag 'overall-evaluation 総合評価', '', class: 'col-sm-3 control-label'
  .col-sm-6
    = select_tag :overall_evaluation, options_for_select((1..5).map { |i| [i
      , i] }, selected: "3" ), class: 'form-control'
.form-group.mt
  = label_tag 'speech', 言葉遣い'', class: 'col-sm-3 control-label'
  .col-sm-6
    = select_tag :speech, options_for_select((1..5).map { |i| [i, i] },
      selected: "3" ), class: 'form-control'

.form-group.mt
  = label_tag 'exact-answer', 質問の答えの正確さ
    '', class: 'col-sm-3 control-label'
  .col-sm-6
    = select_tag :exact_answer, options_for_select((1..5).map { |i| [i, i]
      }, selected: "3" ), class: 'form-control'

.form-group.mt
  = label_tag 'communication', 対人能力'', class: 'col-sm-3 control-label'
  .col-sm-6
    = select_tag :communication, options_for_select((1..5).map { |i| [i, i]
      }, selected: "3" ), class: 'form-control'

.form-group.mt
  = label_tag 'knowledge', 知識'', class: 'col-sm-3 control-label'
  .col-sm-6
    = select_tag :knowledge, options_for_select((1..5).map { |i| [i, i] },
      selected: "3" ), class: 'form-control'

.form-group.mt
  = label_tag 'expression', 表現力'', class: 'col-sm-3 control-label'
  .col-sm-6
    = select_tag :expression, options_for_select((1..5).map { |i| [i, i] },
      selected: "3" ), class: 'form-control'

.form-group.mt

```

```

= label_tag 'impression', 印象'', class: 'col-sm-3 control-label'
.col-sm-6
  = select_tag :impression, options_for_select((1..5).map { |i| [i, i] },
    selected: "3" ), class: 'form-control'

.form-group.mt
= label_tag 'attractive', 魅力'', class: 'col-sm-3 control-label'
.col-sm-6
  = select_tag :attractive, options_for_select((1..5).map { |i| [i, i] },
    selected: "3" ), class: 'form-control'

.form-group.mt
= label_tag 'wish', 志望の強さ'', class: 'col-sm-3 control-label'
.col-sm-6
  = select_tag :wish, options_for_select((1..5).map { |i| [i, i] },
    selected: "3" ), class: 'form-control'

.form-group.mt
= label_tag 'comment', 面接に対するコメント'', class: 'col-sm-3 control-label'
.col-sm-6
  = text_area_tag :comment, '', class: 'form-control', size: "5x10",
    maxlength: '255'
  p
  |
  small 必須ではありません。文字以内のコメントでよろしくお願ひします。 255

.row
  .res_center
    = hidden_field_tag :video_id, @video_id
    = submit_tag 評価を送信'', id: 'send_review', class: "btn btn-lg btn-
      primary", data: { confirm: "評価を送信しますか？ 一度送信すると変更ができません
      1" }

```

9.3.6 ActivationMailer

Listing 25 認可用のメールの本文 (activation_user.text.erb)

<%= @user.name %>様へようこそ。

Interviewのアカウントの仮登録は成功しました。

Interviewユーザ名は

<%= @user.user_id%> です。まだ仮登録のため下記のにアクセスしていただくことでログインが可能となります。

URL

<%= @activation_url %>のご協力ありがとうございます。

Interview

Listing 26 動画を記録した際に全ユーザに通知するメールの本文 (notification_uploaded.text.erb)

からのお知らせです。

Interview 動画が投稿されました。タイトル

: <%= @mail_body_information[: video_title] %>出演者

: <%= @mail_body_information[: cast] %>

URL: <%= @mail_body_information[: video_url] %>評価のほうよろしくお願ひします。いつ
ものご協力ありがとうございます。

Interview

9.3.7 Error

Listing 27 404 ページ (404.html)

```
<!DOCTYPE html>
<html>
<head>
  <title>The page you were looking for doesn't exist (404)</title>
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <style>
    body {
      background-color: #EFEFEF;
      color: #2E2F30;
      text-align: center;
      font-family: arial, sans-serif;
      margin: 0;
    }

    div.dialog {
      width: 95%;
      max-width: 33em;
      margin: 4em auto 0;
    }

    div.dialog > div {
      border: 1px solid #CCC;
      border-right-color: #999;
      border-left-color: #999;
      border-bottom-color: #BBB;
      border-top: #B00100 solid 4px;
      border-top-left-radius: 9px;
      border-top-right-radius: 9px;
      background-color: white;
      padding: 7px 12% 0;
      box-shadow: 0 3px 8px rgba(50, 50, 50, 0.17);
    }

    h1 {
      font-size: 100%;
      color: #730E15;
      line-height: 1.5em;
    }
  </style>
</head>
</html>
```

```

}

div.dialog > p {
  margin: 0 0 1em;
  padding: 1em;
  background-color: #F7F7F7;
  border: 1px solid #CCC;
  border-right-color: #999;
  border-left-color: #999;
  border-bottom-color: #999;
  border-bottom-left-radius: 4px;
  border-bottom-right-radius: 4px;
  border-top-color: #DADADA;
  color: #666;
  box-shadow: 0 3px 8px rgba(50, 50, 50, 0.17);
}
</style>
</head>

<body>
<!-- This file lives in public/404.html -->
<div class="dialog">
  <div>
    <h1>The page you were looking for doesn't exist.</h1>
    <p>You may have mistyped the address or the page may have moved.</p>
  </div>
  <p>If you are the application owner check the logs for more information.</p>
</div>
</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
  <title>The change you wanted was rejected (422)</title>
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <style>
body {
  background-color: #EFEFEF;
  color: #2E2F30;
  text-align: center;
  font-family: arial, sans-serif;
  margin: 0;
}

div.dialog {
  width: 95%;
  max-width: 33em;
  margin: 4em auto 0;
}

div.dialog > div {
  border: 1px solid #CCC;
  border-right-color: #999;
  border-left-color: #999;
  border-bottom-color: #BBB;
  border-top: #B00100 solid 4px;
  border-top-left-radius: 9px;
  border-top-right-radius: 9px;
  background-color: white;
  padding: 7px 12% 0;
  box-shadow: 0 3px 8px rgba(50, 50, 50, 0.17);
}

h1 {
  font-size: 100%;
  color: #730E15;
  line-height: 1.5em;
}

```

```

div.dialog > p {
  margin: 0 0 1em;
  padding: 1em;
  background-color: #F7F7F7;
  border: 1px solid #CCC;
  border-right-color: #999;
  border-left-color: #999;
  border-bottom-color: #999;
  border-bottom-left-radius: 4px;
  border-bottom-right-radius: 4px;
  border-top-color: #DADADA;
  color: #666;
  box-shadow: 0 3px 8px rgba(50, 50, 50, 0.17);
}
</style>
</head>

<body>
<!-- This file lives in public/422.html -->
<div class="dialog">
  <div>
    <h1>The change you wanted was rejected.</h1>
    <p>Maybe you tried to change something you didn't have access to.</p>
  </div>
  <p>If you are the application owner check the logs for more information.</p>
</div>
</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
  <title>We're sorry, but something went wrong (500)</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
body {
  background-color: #EFEFEF;
  color: #2E2F30;
  text-align: center;
  font-family: arial, sans-serif;
  margin: 0;
}

div.dialog {
  width: 95%;
  max-width: 33em;
  margin: 4em auto 0;
}

div.dialog > div {
  border: 1px solid #CCC;
  border-right-color: #999;
  border-left-color: #999;
  border-bottom-color: #BBB;
  border-top: #B00100 solid 4px;
  border-top-left-radius: 9px;
  border-top-right-radius: 9px;
  background-color: white;
  padding: 7px 12% 0;
  box-shadow: 0 3px 8px rgba(50, 50, 50, 0.17);
}

h1 {
  font-size: 100%;
  color: #730E15;
  line-height: 1.5em;
}

```

```

div.dialog > p {
  margin: 0 0 1em;
  padding: 1em;
  background-color: #F7F7F7;
  border: 1px solid #CCC;
  border-right-color: #999;
  border-left-color: #999;
  border-bottom-color: #999;
  border-bottom-left-radius: 4px;
  border-bottom-right-radius: 4px;
  border-top-color: #DADADA;
  color: #666;
  box-shadow: 0 3px 8px rgba(50, 50, 50, 0.17);
}
</style>
</head>

<body>
  <!-- This file lives in public/500.html -->
  <div class="dialog">
    <div>
      <h1>We're sorry, but something went wrong.</h1>
    </div>
    <p>If you are the application owner check the logs for more information.</p>
  </div>
</body>
</html>

```

9.4 Mailers

Listing 30 メールの設定 (application_mailer.rb)

```
class ApplicationMailer < ActionMailer::Base
  default from: "andrewlabtest@gmail.com"
end
```

Listing 31 認可用のメールの処理 (activation_mailer.rb)

```
class ActivationMailer < ApplicationMailer
  def activation_user(user_email)
    @user = User.find_by(email: user_email)
    case Rails.env
    when 'development' || 'test'
      @activation_url = "interview.com:3000/user/activation/#{@user.
        activation_digest}"
    when 'production'
      @activation_url = "http://koudai.a4w.ise.osaka-sandai.ac.jp/user/
        activation/#{@user.activation_digest}"
    end
    mail(to: @user.email, subject: 'システムの本登録のお知らせinterview')
  end
end
```

Listing 32 動画を記録した際のメールの処理 (video_mailer.rb)

```
class VideoMailer < ApplicationMailer
  def notification_uploaded(user_id:, video_title:, video_id:, email:)
    @mail_body_information = mail_body(user_id: user_id, video_title:
      video_title, video_id: video_id)
    mail(to: email, subject: 'interview_動画が投稿されました')
  end

  def cast_name(cast_id)
    User.where(id: cast_id).first.name
  end

  def video_url(video_id)
    case Rails.env
    when 'development' || 'test'
      "interview.com:3000/videos/#{video_id}"
    when 'production'
      "http://koudai.a4w.ise.osaka-sandai.ac.jp/videos/#{video_id}"
    end
  end

  def mail_body(user_id:, video_title:, video_id:)
    {
      cast: cast_name(user_id),
      video_title: video_title,
      video_url: video_url(video_id),
    }
  end
end
```

9.5 Helper

Listing 33 SessionHelper (sessions_helper.rb)

```
module SessionsHelperログイン系
  #
  def log_in(user)
    session[:user_id] = user.id
  end

  def remember(user)
    user.remember
    cookies.permanent.signed[:user_id] = user.id
    cookies.permanent[:remember_token] = user.remember_token
  end

  def current_user
    if (user_id = session[:user_id])
      @current_user ||= User.find_by(id: user_id)
    elsif (user_id = cookies.signed[:user_id])
      user = User.find_by(id: user_id)
      if user && user.authenticated?(cookies[:remember_token])
        log_in user
        @current_user = user
      end
    end
  end

  def current_youtube_user
    @current_youtube_user ||= YoutubeUser.find_by(id: session[:user_id]) if
      session[:user_id]
  end

  def logged_in?
    !current_user.nil?
  endログアウト系
  #

  def forget(user)
    user.forget
  end
end
```

```
cookies.delete(:user_id)
cookies.delete(:remember_token)
end

def log_out
  forget(current_user)
  session.delete(:user_id)
  @current_user = nil
end
end
```

Listing 34 VideosHelper (videos_helper.rb)

```
module VideosHelper
  def current_youtube_user
    @current_youtube_user = YoutubeUser.find_by(name: 'labo_test')
  end

  def get_user_list
    User.select(:id, :name).where(activated: true)
  end

  def reviewed?(user_id, video_id)
    Review.reviewed?(user_id, video_id)
  end
end
```

9.6 Assets

9.6.1 JavaScripts

Listing 35 JavaScript の読み込みリスト (application.js)

```
// This is a manifest file that'll be compiled into application.js, which will
  include all the files
// listed below.
//
// Any JavaScript/Coffee file within this directory, lib/assets/javascripts,
  vendor/assets/javascripts,
// or any plugin's vendor/assets/javascripts directory can be referenced here
  using a relative path.
//
// It's not advisable to add code directly here, but if you do, it'll appear
  at the bottom of the
// compiled file.
//
// Read Sprockets README (https://github.com/rails/sprockets#sprockets-
  directives) for details
// about supported directives.
//
//
//= require jquery
//= require jquery_ujs
//= require twitter/bootstrap
//= require turbolinks
//= require_tree .
```

9.6.2 Stylesheets

Listing 36 CSS の読み込みリスト (application.css)

```
/*
 * This is a manifest file that'll automatically include all the stylesheets
 * available in this directory
 * and any sub-directories. You're free to add application-wide styles to this
 * file and they'll appear at
 * the top of the compiled file, but it's generally better to create a new
 * file per style scope.
 *= require_self
 *= require_tree .
*/
```

Listing 37 Bootstrap のカスタム CSS (custom.css.scss)

```
/*
@import "bootstrap-sprockets";
@import "bootstrap";
*/

/* color */
$light-gray: #777;

/* universal */

body{
  padding-top: 0px;
}

section{
  overflow: auto;
}

textarea{
  resize: vertical;
}

.center{
  text-align: center;
  h1 {
    margin-bottom: 10px;
  }
}

/* typography */

h1, h2, h3, h4, h5, h6{
  line-height: 1;
}

h1{
  font-size: 3em;
  letter-spacing: -2px;
}
```

```

margin-bottom: 30px;
text-align: center;
}

h2{
font-size: 1.2em;
letter-spacing: -1px;
margin-bottom: 30px;
text-align: center;
font-weight: normal;
color: $light-gray;
}

p{
font-size: 1.1em;
line-height: 1.7em;
}

/* footer */

footer{
margin-top: 45px;
padding-top: 5px;
border-top: 1px solid #eaeaea;
color: $light-gray;
footer a{
color: #555;
&:hover{
color: #222;
}
}
small{
float: left;
}
ul{
float: right;
list-style: none;
li{
float: left;
margin-left: 15px;
}
}

```

```

    }
  }
}

/*の調整gridgutter*/

.row-0{
  margin-left:0px;
  margin-right:0px;

  > div{
    padding-right:0px;
    padding-left:0px;
  }
}

.row-10{
  margin-left:-5px;
  margin-right:-5px;

  > div{
    padding-right:5px;
    padding-left:5px;
  }
}

.row-20{
  margin-left:-10px;
  margin-right:-10px;

  > div{
    padding-right:10px;
    padding-left:10px;
  }
}

.row-30{
  margin-left:-15px;
  margin-right:-15px;

  > div{
    padding-right:15px;
    padding-left:15px;
  }
}

```

```

    }
}
.row-40{
    margin-left:-20px;
    margin-right:-20px;

    > div{
        padding-right:20px;
        padding-left:20px;
    }
}

.navbar{
    border-radius: 0;
    border: none;
    background-color: #003355;
}

.navbar-default .navbar-nav >li > a{
    color: #fff;
}

.navbar-default .navbar-nav >li > a:hover{
    color: #aaa;
}

```

Listing 38 Sessions の CSS (sessions.css.scss)

```
h1{  
  text-align: center;  
}
```

Listing 39 Users の CSS (users.css.scss)

```
.res_center {
  display: table;
  margin: 0 auto;
}

.video-detail {
  width: 400px;
  margin: 0 auto;
}

.movie-wrap {
  position: relative;
  padding-bottom: 56.25%;
  padding-top: 30px;
  height: 0;
  overflow: hidden;
}

.movie-wrap iframe {
  position: absolute;
  top: 0;
  width: 100%;
  height: 100%;
}

.comment {
  position: relative;
  width: 100%;
  height: 100%;
  margin-top: 30px;
  margin-right: 40px;
  top: 50px;
}
```

Listing 40 Videos の CSS (videos.css.scss)

```
#contents{
  position: relative;
  width: 1170px;
  height: 700px;
  margin: 0 auto;
  padding-top: 30px;
}

#movie{
  margin: 0 auto;
  width: 600px;
  height: 600px;
}

.movie-wrap{
  position: relative;
  padding-bottom: 56.25%;
  padding-top: 30px;
  height: 0;
  overflow: hidden;
}

.movie-wrap iframe {
  position: absolute;
  top: 0;
  width: 100%;
  height: 100%;
}

.form{
  position: relative;
  width: 500px;
  height: 600px;
  margin: 0 auto;
  margin-top: 30px;
  margin-right: 40px;
  top: 50px;
}
```

```
.preloader {
  display: none;
}

.mt {
  margin-top: 20px;
}

.center{
  margin: 0 auto;
}

.res_center {
  display: table;
  margin: 0 auto;
}
```

9.7 Library

Listing 41 YouTube に動画をアップロードする処理 (video_upload.rb)

```
class VideoUpload

  require 'net/http'
  require 'uri'
  require 'google/api_client'
  require 'json'

  attr_reader :video_id, :video_content_type

  def initialize(params)
    @auth_token = params[:auth_token]
    @video_size = params[:video_size]
    @video_title = params[:video_title]
    @video_content_type = params[:video_content_type]
    @upload_url = nil
    @video_id = nil
  end

  def setup!
    session_req = session_youtube
    if valid_token?(session_req)
      store_upload_url(session_req)
      true
    else
      false
    end
  end

  def start!(video_file)
    upload_res = upload_to_youtube(video_file)
    upload_result = JSON.parse(upload_res.body)
    if uploaded?(upload_result)
      @video_id = upload_result["id"]
    else
      false
    end
  end
end
```

```

end

private

def session_youtube
  uri = URI.parse(Constants::Youtube::APIURL)
  puts "host_#{uri.host}_port_#{uri.port}_request_uri_#{uri.request_uri}"
  https = Net::HTTP.new(uri.host, uri.port)
  https.use_ssl = true
  params_req = Net::HTTP::Post.new(uri.request_uri)
  params_req['Authorization'] = "Bearer_#{@auth_token}"
  params_req.body = create_session_body.to_json
  params_req['Content-Length'] = params_req.body.length
  params_req['Content-Type'] = 'application/json; charset=UTF-8'
  params_req['X-Upload-Content-Length'] = @video_size
  params_req['X-Upload-Content-Type'] = @video_content_type
  res = https.request(params_req)
end

def create_session_body
  body = {
    snippet:{
      title: @video_title,
      description: "面接用",
      tags: [],
    },
    status:{
      privacyStatus: "unlisted",
      embeddable: 'True',
      license: "youtube"
    }
  }
end

def valid_token?(target)
  target.each_value do |value|
    return false if value =~ /invalid_token$/
  end
end
end

```

```

def store_upload_url(target)
  target.each do |key, value|
    if key =~ /location/
      @upload_url = value
      break
    end
  end
end

def upload_to_youtube(video_file)
  uri = URI.parse(@upload_url)
  https = Net::HTTP.new(uri.host, uri.port)
  https.use_ssl = true
  params_req = Net::HTTP::Put.new(uri.request_uri)
  params_req['Authorization'] = "Bearer #{@auth_token}"
  params_req['Content-Length'] = @video_size
  params_req['Content-Type'] = @video_content_type
  params_req.body = video_file
  res = https.request(params_req)
end

def uploaded?(result)
  if result["status"]["uploadStatus"] == 'uploaded'
    true
  else
    false
  end
end
end

```

9.8 Task

Listing 42 Unicorn のタスク (unicorn.rake)

```
namespace :unicorn do
  desc "Start unicorn development env."
  task(:start){
    config = Rails.root.join('config', 'unicorn.rb')
    sh "bundle_exec_unicorn_rails -c#{config} -E development -D"
  }

  desc "Stop unicorn"
  task(:stop) { unicorn_signal :QUIT }

  desc "Restart unicorn withUSR2"
  task(:restart) { unicorn_signal :USR2 }

  def unicorn_signal signal
    Process.kill signal, unicorn_pid
  end

  def unicorn_pid
    begin
      File.read("../tmp/unicorn.pid").to_i
    rescue Errno::ENOENT
      raise "Unicorn doesn't seem to be running"
    end
  end
end
```

9.9 Config

Listing 43 rackup ファイル (config.ru)

```
# This file is used by Rack-based servers to start the application.

require 'unicorn/worker_killer'
use Unicorn::WorkerKiller::MaxRequests, 100, 200, true
use Unicorn::WorkerKiller::Oom, (192*(1024**2)), (256*(1024**2)), 16

require ::File.expand_path('../config/environment', __FILE__)
run Rails.application
```

Listing 44 rake の設定ファイル (Rakefile)

```
# Add your own tasks in files placed in lib/tasks ending in .rake,  
# for example lib/tasks/capistrano.rake, and they will automatically be  
available to Rake.
```

```
require File.expand_path('../config/application', __FILE__)
```

```
Rails.application.load_tasks
```

Listing 45 Gemfile (Gemfile)

```

source 'https://rubygems.org'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '4.2.6'
# Use postgresql as the database for Active Record
gem 'pg', '~>_0.15'
# Use SCSS for stylesheets
gem 'sass-rails', '~>_5.0'
# Use Uglifier as compressor for JavaScript assets
gem 'uglifier', '>=_1.3.0'
# Use CoffeeScript for .coffee assets and views
gem 'coffee-rails', '~>_4.1.0'
# See https://github.com/rails/execjs#readme for more supported runtimes
gem 'therubyracer', platforms: :ruby

# Use jquery as the JavaScript library
gem 'jquery-rails'
# Turbolinks makes following links in your web application faster. Read more:
  https://github.com/rails/turbolinks
gem 'turbolinks'
# Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
gem 'jbuilder', '~>_2.0'
# bundle exec rake doc:rails generates the API under doc/api.
gem 'sdoc', '~>_0.4.0', group: :doc

# Use ActiveSupport has_secure_password
gem 'bcrypt', '~>_3.1.7'

# Use Unicorn as the app server
gem 'unicorn'
gem 'unicorn-worker-killer'

# Use Capistrano for deployment
# gem 'capistrano-rails', group: :development

gem 'less-rails'
gem 'twitter-bootstrap-rails'
gem 'autoprefixer-rails'

```

```

gem 'slim-rails'

gem 'omniauth'
gem 'omniauth-google-oauth2'
gem 'omniauth-youtube'

gem 'google-api-client', '~>0.8.2', require: 'google/api_client'

gem 'jquery-turbolinks'

# ページャー用
gem 'kaminari'

# に変数を渡す用 coffeescript
gem 'gon'

#用のグラフ表示用 mypage
gem 'chartkick'

gem 'mimemagic'

group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger
  console
  gem 'byebug'
  gem 'rspec-rails'
  gem 'factory_girl_rails'
end

group :development do
  # Access an IRB console on exception pages or by using <%= console %> in
  views
  gem 'web-console', '~>_2.0'

  # Spring speeds up development by keeping your application running in the
  background. Read more: https://github.com/rails/spring
  gem 'spring'
end

group :test do

```

```
gem 'faker '  
gem 'capybara '  
gem 'database_cleaner '  
gem 'launchy '  
gem 'selenium-webdriver '  
end
```

Listing 46 Ruby on Rails の設定 (application.rb)

```

require File.expand_path('../boot', __FILE__)

require "rails"
# Pick the frameworks you want:
require "active_model/railtie"
require "active_job/railtie"
require "active_record/railtie"
require "action_controller/railtie"
require "action_mailer/railtie"
require "action_view/railtie"
require "sprockets/railtie"
# require "rails/test_unit/railtie"

# Require the gems listed in Gemfile, including any gems
# you've limited to :test, :development, or :production.
Bundler.require(*Rails.groups)

module Interview
  class Application < Rails::Application
    # Settings in config/environments/* take precedence over those specified
    # here.
    # Application configuration should go into files in config/initializers
    # — all .rb files in that directory are automatically loaded.

    # Set Time.zone default to the specified zone and make Active Record auto-
    # convert to this zone.
    # Run "rake -D time" for a list of tasks for finding time zone names.
    # Default is UTC.
    # config.time_zone = 'Central Time (US & Canada)'

    # The default locale is :en and all translations from config/locales/*.rb,
    # yml are auto loaded.
    # config.i18n.load_path += Dir[Rails.root.join('my', 'locales', '*.{rb,yml}
    #   }').to_s]
    # config.i18n.default_locale = :de

    # Do not swallow errors in after_commit/after_rollback callbacks.
    config.active_record.raise_in_transactional_callbacks = true
  end
end

```

```
# /lib autoload
config.autoload_paths += %W(#{config.root}/lib)

config.generators do |g|
  g.test_framework :rspec,
    fixtures: true,
    view_spacers: false,
    helper_specs: false,
    routing_specs: false,
    controller_specs: true,
    request_specs: false
  g.fixture_replacement :factory_girl, dir: "spec/factories"
end
end
end
```

Listing 47 データベースの設定ファイル (database.yml)

```

# PostgreSQL. Versions 8.2 and up are supported.
#
# Install the pg driver:
#   gem install pg
# On OS X with Homebrew:
#   gem install pg -- --with-pg-config=/usr/local/bin/pg-config
# On OS X with MacPorts:
#   gem install pg -- --with-pg-config=/opt/local/lib/postgresql84/bin/
#     pg-config
# On Windows:
#   gem install pg
#     Choose the win32 build.
#     Install PostgreSQL and put its /bin directory on your path.
#
# Configure Using Gemfile
# gem 'pg'
#
default: &default
  adapter: postgresql
  encoding: utf8
  # For details on connection pooling, see rails configuration guide
  # http://guides.rubyonrails.org/configuring.html#database-pooling
  pool: 5

development:
  <<: *default
  database: interview_development

  # The specified database role being used to connect to postgres.
  # To create additional roles in postgres see '$ createuser --help '.
  # When left blank, postgres will use the default role. This is
  # the same name as the operating system user that initialized the database.
  username:

  # The password associated with the postgres role (username).
  password:

  # Connect on a TCP socket. Omitted by default since the client uses a
  # domain socket that doesn't need configuration. Windows does not have

```

```

# domain sockets, so uncomment these lines.
host: localhost

# The TCP port the server listens on. Defaults to 5432.
# If your server runs on a different port number, change accordingly.
port: 5432

# Schema search path. The server defaults to $user,public
#schema_search_path: myapp,sharedapp,public

# Minimum log levels, in increasing order:
#  debug5, debug4, debug3, debug2, debug1,
#  log, notice, warning, error, fatal, and panic
# Defaults to warning.
#min_messages: notice

# Warning: The database defined as "test" will be erased and
# re-generated from your development database when you run "rake".
# Do not set this db to the same as development or production.
test:
  <<: *default
  database: interview_test
  username:
  password:
  host:
  port: 5432
production:
  <<: *default
  database: interview_production
  username:
  password:
  host:
  port: 5432

```

Listing 48 環境の設定ファイル (environment.rb)

```
# Load the Rails application.  
require File.expand_path('../application', __FILE__)  
  
# Initialize the Rails application.  
Rails.application.initialize!
```

Listing 49 パスの定義ファイル (routes.rb)

```
Rails.application.routes.draw do

  root to: 'static_pages#home'
  get 'static_pages/help'

  get 'login' => 'sessions#new'
  post 'login' => 'sessions#create'
  delete 'logout' => 'sessions#destroy'

  get 'user/score' => 'users#score', as: :user_score
  get 'user/videos' => 'users#video_list', as: :user_video
  get 'user/videos/:id' => 'users#show', as: :user_show
  get 'user/registration' => 'users#new'
  post 'user/registration' => 'users#create', as: :create_user
  get 'user/activation/:id' => 'users#activation'

  get '/auth/:provider/callback' => 'auths#create'
  get '/auth/failure' => 'auths#failure'

  get '/videos/upload' => 'videos#new', as: :new_upload
  post '/videos/upload' => 'videos#upload'

  resources :videos, only: [ :index, :show]

  post '/videos/show' => 'reviews#create', as: :create_review

end
```

Listing 50 Unicorn の設定ファイル (unicorn.rb)

```

worker_processes Integer(ENV["WEB_CONCURRENCY"] || 1)
timeout 1000
preload_app true

rails_root = File.expand_path('../../', __FILE__)

listen "#{rails_root}/tmp/unicorn.sock"
pid "#{rails_root}/tmp/unicorn.pid"

before_fork do |server, worker|
  Signal.trap 'TERM' do
    puts 'Unicorn master intercepting TERM and sending myself QUIT instead'
    Process.kill 'QUIT', Process.pid
  end

  defined?(ActiveRecord::Base) and
    ActiveRecord::Base.connection.disconnect!
end

after_fork do |server, worker|
  Signal.trap 'TERM' do
    puts 'Unicorn worker intercepting TERM and doing nothing. Wait for master to send QUIT'
  end

  defined?(ActiveRecord::Base) and
    ActiveRecord::Base.establish_connection
end

# ログの出力
stderr_path File.expand_path('log/unicorn.log', ENV['RAILS_ROOT'])
stdout_path File.expand_path('log/unicorn.log', ENV['RAILS_ROOT'])

```

```

Rails.application.configure do
  # Settings specified here will take precedence over those in config/
  # application.rb.

  # In the development environment your application's code is reloaded on
  # every request. This slows down response time but is perfect for
  # development
  # since you don't have to restart the web server when you make code changes.
  config.cache_classes = false

  # Do not eager load code on boot.
  config.eager_load = false

  # Show full error reports and disable caching.
  config.consider_all_requests_local = true
  config.action_controller.perform_caching = false

  # Don't care if the mailer can't send.
  config.action_mailer.raise_delivery_errors = false

  # Print deprecation notices to the Rails logger.
  config.active_support.deprecation = :log

  # Raise an error on page load if there are pending migrations.
  config.active_record.migration_error = :page_load

  # Debug mode disables concatenation and preprocessing of assets.
  # This option may cause significant delays in view rendering with a large
  # number of complex assets.
  config.assets.debug = true

  # Asset digests allow you to set far-future HTTP expiration dates on all
  # assets,
  # yet still be able to expire them through the digest params.
  config.assets.digest = true

  # Adds additional error checking when serving assets at runtime.
  # Checks for improperly declared sprockets dependencies.
  # Raises helpful error messages.

```

```
config.assets.raise_runtime_errors = true

# Raises error for missing translations
# config.action_view.raise_on_missing_translations = true

# activation_mailer
config.action_mailer.delivery_method = :smtp
config.action_mailer.raise_delivery_errors = true
config.action_mailer.smtp_settings = {
  address: 'smtp.gmail.com',
  port: 587,
  domain: 'interview.com:3000',
  user_name: 'andrewlabotest@gmail.com',
  password: ENV['email_password'],
  enable_starttls_auto: true,
  authentication: :plain
}
end
```

Listing 52 test 環境の設定 (test.rb)

```

Rails.application.configure do
  # Settings specified here will take precedence over those in config/
  # application.rb.

  # The test environment is used exclusively to run your application's
  # test suite. You never need to work with it otherwise. Remember that
  # your test database is "scratch space" for the test suite and is wiped
  # and recreated between test runs. Don't rely on the data there!
  config.cache_classes = true

  # Do not eager load code on boot. This avoids loading your whole application
  # just for the purpose of running a single test. If you are using a tool
  # that
  # preloads Rails for running tests, you may have to set it to true.
  config.eager_load = false

  # Configure static file server for tests with Cache-Control for performance.
  config.serve_static_files = true
  config.static_cache_control = 'public, _max-age=3600'

  # Show full error reports and disable caching.
  config.consider_all_requests_local = true
  config.action_controller.perform_caching = false

  # Raise exceptions instead of rendering exception templates.
  config.action_dispatch.show_exceptions = false

  # Disable request forgery protection in test environment.
  config.action_controller.allow_forgery_protection = false

  # Tell Action Mailer not to deliver emails to the real world.
  # The :test delivery method accumulates sent emails in the
  # ActionMailer::Base.deliveries array.
  config.action_mailer.delivery_method = :test

  # Randomize the order test cases are executed.
  config.active_support.test_order = :random

  # Print deprecation notices to the stderr.

```

```
config.active_support.deprecation = :stderr

# Raises error for missing translations
# config.action_view.raise_on_missing_translations = true
end
```

```

Rails.application.configure do
  # Settings specified here will take precedence over those in config/
  # application.rb.

  # Code is not reloaded between requests.
  config.cache_classes = true

  # Eager load code on boot. This eager loads most of Rails and
  # your application in memory, allowing both threaded web servers
  # and those relying on copy on write to perform better.
  # Rake tasks automatically ignore this option for performance.
  config.eager_load = true

  # Full error reports are disabled and caching is turned on.
  config.consider_all_requests_local = false
  config.action_controller.perform_caching = true

  # Enable Rack::Cache to put a simple HTTP cache in front of your application
  # Add 'rack-cache' to your Gemfile before enabling this.
  # For large-scale production use, consider using a caching reverse proxy
  # like
  # NGINX, varnish or squid.
  # config.action_dispatch.rack_cache = true

  # Disable serving static files from the '/public' folder by default since
  # Apache or NGINX already handles this.
  config.serve_static_files = true

  # Compress JavaScripts and CSS.
  config.assets.js_compressor = :uglifier
  # config.assets.css_compressor = :sass

  # Do not fallback to assets pipeline if a precompiled asset is missed.
  config.assets.compile = false

  # Asset digests allow you to set far-future HTTP expiration dates on all
  # assets,
  # yet still be able to expire them through the digest params.
  config.assets.digest = true

```

```

# 'config.assets.precompile' and 'config.assets.version' have moved to
  config/initializers/assets.rb

# Specifies the header that your server uses for sending files.
# config.action_dispatch.x_sendfile_header = 'X-Sendfile' # for Apache
# config.action_dispatch.x_sendfile_header = 'X-Accel-Redirect' # for NGINX

# Force all access to the app over SSL, use Strict-Transport-Security, and
  use secure cookies.
# config.force_ssl = true

# Use the lowest log level to ensure availability of diagnostic information
# when problems arise.
config.log_level = :debug

# Prepend all log lines with the following tags.
# config.log_tags = [ :subdomain, :uuid ]

# Use a different logger for distributed setups.
# config.logger = ActiveSupport::TaggedLogging.new(SyslogLogger.new)

# Use a different cache store in production.
# config.cache_store = :mem_cache_store

# Enable serving of images, stylesheets, and JavaScripts from an asset
  server.
# config.action_controller.asset_host = 'http://assets.example.com'

# Ignore bad email addresses and do not raise email delivery errors.
# Set this to true and configure the email server for immediate delivery to
  raise delivery errors.
# config.action_mailer.raise_delivery_errors = false

# Enable locale fallbacks for I18n (makes lookups for any locale fall back
  to
# the I18n.default_locale when a translation cannot be found).
config.i18n.fallbacks = true

# Send deprecation notices to registered listeners.

```

```

config.active_support.deprecation = :notify

# Use default logging formatter so that PID and timestamp are not suppressed
.
config.log_formatter = ::Logger::Formatter.new

# Do not dump schema after migrations.
config.active_record.dump_schema_after_migration = false

config.action_mailer.delivery_method = :smtp
config.action_mailer.raise_delivery_errors = true
config.action_mailer.smtp_settings = {
  address: 'smtp.gmail.com',
  port: 587,
  domain: 'koudai.a4w.ise.osaka-sandai.ac.jp',
  user_name: 'andrewlabotest@gmail.com',
  password: ENV['email_password'],
  enable_starttls_auto: true,
  authentication: :plain,
  openssl_verify_mode: 'none'
}
end

```