

修 士 論 文

令和五年度

近畿大学大学院

総合理工学研究科

エレクトロニクス系工学専攻

21-3-334-0415 番 栗 岡 陽 平

修 士 論 文

令和五年度

論文内容の要旨

グラフデータベースを用いた
学習者理解度可視化システムの開発

近畿大学大学院

総合理工学研究科

エレクトロニクス系工学専攻

21-3-334-0415 番 栗 岡 陽 平

2019 年 12 月に文部科学省が作成した「教育の情報化に関する手引き」[1]によると教育の情報化が促進されている。e ラーニング [2] は「情報通信技術の時間的・空間的制約をなくす」、「双方向性を有する」、「カスタマイズを容易にする」という特性を有するシステムのうちの一つであることから、教育の情報化に有効である。e ラーニング上で学習するにあたり、自身の学習目標を設定することは、学びを深める手段のうちの一つである [3]。一方、学習目標を設定するには、自身が学習したい対象の知識を把握している必要がある。しかし、学習者自身では学習項目を理解していると主観的には考えていても、他人が客観的に判断すると理解できていない場合があり、学習者自身で学習目標を設定することは必ずしも容易ではない。

東本氏らの研究では、科学領域においては習得すべきさまざまな概念および概念間の関係が存在し、その一つに概念の階層構造を学習者に理解させることは科学の学習において重要な課題であると認識していた。そこで、階層構造の理解の促進を目的とした学習者自身によるコンセプトマップ (以下、CMap)[4] の構築のためのシステムを開発した [5]。野村氏らの研究では、学習方法の一つとして、学習した内容を整理して他の学習者に教える事で自信の理解を深める方法があり、他者に対して学習内容を理解させることができるか否かで自身の理解が十分であるか否かを学習者自身が再確認することができるという教え合い学習をシステムの推奨を用いて実際に学習者間で行わせることを目的とした研究を進めている。[6] 平塚氏らの研究では、高等教育機関における学生たちに対して、教育課程を理解してもらうことが重要であると考え、教務システムと e ポートフォリオを連携した「学習成果可視化システム」を構築した。このシステムはオープンソースのシステムを用いて構築し、公開・フィードバックする点で意義があるとしている。[7]

本研究では、学習目標の設定支援を目的に、学習者の理解度を可視化する、グラフデータベースを用いた学習者理解度可視化システムを開発した。本システムは e ラーニングで学習している学習者を対象としたシステムで、CMap を利用して学習者が学習目標を設定する場合に本システムを利用することを想定している。学習者は指導者が作成した問題を解き、本システムを用いて CMap を作成する。本システムでは学習者の回答情報から CMap を作成し、学習者は自身が作成した CMap と、システムが生成した CMap を比較することにより、自身の学習理解度を客観的に確認でき、学習目標設定の基準にできる。

CMap を作成するにおいて、事前に学習目標と学習項目の情報が必要になる。そこで本システムでは、CMap をキットビルド概念マップシステム [8] を用いて作成しており、事前に学習者を指導する指導者が本システムを用いて CMap に関する学習目標、学習項目を入力できる。これにより、学習者は事前に設定された学習目標、学習項目を基に本システムを用いて自身で CMap を作成できる。

本システムには、グラフデータ管理機能、グラフデータ入力補助機能、グラフデータ可視化機能が存在し、グラフデータ管理機能では、グラフデータベースの内の一つである Neo4j を用いてグラフデータを管理し、学習者・指導者にグラフデータを通して CMap の学習目標、学習項目を提供する。グラフデータ入力補助機能は指導者が CMap の学習目標、学習項目を入力するときに、そのデータを木構造で入力できるフォームを用意し、そのデータをグラフデータへと変換し Neo4j に登録する機能である。グラフデータ可視化機能は、学習者に対して指導者が入力した CMap の

グラフデータを可視化し提供する機能である。また、学習者が CMap を作成するためのフォームも提供している。

本研究では、グラフデータ可視化機能を用いて学習者に座学における学習目標設定方法と本システムを用いた学習目標設定方法を比較するための確認テストによる評価実験を実施した。実験の結果、本システム使用者と本システム非利用者の学習目標設定方法では本システム利用者の学習目標設定方法の方が確認テストの結果が向上していることを確認した。

修 士 論 文

グラフデータベースを用いた 学習者理解度可視化システムの開発

Development of a Visualization System for Learner Comprehension
Using a Graph Database

目次

第 1 章	序論	2
1.1	序論	2
第 2 章	使用技術	4
2.1	使用技術	5
2.2	研究内容	14
2.3	実験・考察	36
2.4	結論・今後の課題	38
	謝辞	39
	参考文献	40

第 1 章 序論

1.1 序論

1.1.1 本章の概要

本章では、研究背景と研究目的、評価実験の概要、そして本論文の構成について記載する。
研究背景では、教育の情報化に関する課題、本研究に関する研究について記載している。
研究の目的では、本研究の目的を記載している。
研究の内容では、開発したシステムとシステム各部の機能の説明を記載している。
評価実験の概要では、システムを評価する為実施した評価実験の内容と結果を記載している。
また、本論文の構成では各章を番号付きでリストで記載している。

1.1.2 研究背景

2019 年 12 月に文部科学省が作成した「教育の情報化に関する手引き」[1]によると教育の情報化が促進されている。e ラーニング [2] は「情報通信技術の時間的・空間的制約をなくす」、「双方向性を有する」、「カスタマイズを容易にする」という特性を有するシステムのうちの一つであることから、教育の情報化に有効である。e ラーニング上で学習するにあたり、自身の学習目標を設定することは、学びを深める手段のうちの一つである [3]。

一方、学習目標を設定するには、自身が学習したい対象の知識を把握している必要がある。しかし、学習者自身では学習項目を理解していると主観的には考えていても、他人が客観的に判断すると理解できていない場合があり、学習者自身で学習目標を設定することは必ずしも容易ではない。

東本氏らの研究では、科学領域においては習得すべきさまざまな概念および概念間の関係が存在し、その一つに概念の階層構造を学習者に理解させることは科学の学習において重要な課題であると認識していた。そこで、階層構造の理解の促進を目的とした学習者自身によるコンセプトマップ (以下, CMap)[4] の構築のためのシステムを開発した [5]。

野村氏らの研究では、学習方法の一つとして、学習した内容を整理して他の学習者に教える事で自信の理解を深める方法があり、他者に対して学習内容を理解させることができるか否かで自身の理解が十分であるか否かを学習者自身が再確認することができるという教え合い学習をシステムの推奨を用いて実際に学習者間で行わさせることを目的とした研究を進めている [6]。

平塚氏らの研究では、高等教育機関における学生たちに対して、教育課程を理解してもらうことが重要であると考え、教務システムと e ポートフォリオを連携した「学習成果可視化システム」を

構築した。このシステムはオープンソースのシステムを用いて構築し、公開・フィードバックする点で意義があるとしている [7].

西川氏らの研究では、大学における学生がディプロマポリシーに向けて現段階でどのような学修を積み立てているのか確認することを目的とした、グラフデータベース Neo4j による学習ポートフォリオ作成支援システムを開発している [9]. この研究で、ディプロマポリシーに向けて学修をどのように積み立てているかを可視化でき、それによりディプロマポリシーに向けた学修達成度を把握でき、学生がその後どのように履修計画を立案するかの指標となることが示された。

1.1.3 研究の目的

本研究では、CMap を用いて学習目標を設定する学習者を対象としその学習目標設定の支援を目的としている。

1.1.4 研究の内容

本研究では、グラフデータベースを用いて学習者の理解度を可視化し、学習目標の設定を支援できるグラフデータベースを用いた学習者理解度可視化システム (以下、本システム) を用いた学習者理解度可視化システムを開発した。

学習目標を設定するには自身の学習度合いを正確に把握必要がある。しかし、自身の学習度合いを主観的に把握できても客観的に見ると誤っている可能性がある。そこで本システムのグラフデータ可視化機能により学習者のテストの回答情報と、指導者による学習目標、学習項目の情報からグラフデータベースを用いて CMap を自動的に作成することにより、学習者自身が作成した CMap と本システムが自動的に作成した CMap を比較することにより、客観的に学習者の理解度を把握できる。

グラフデータで CMap を作成するにあたり、本システムにはグラフデータ管理機能とグラフデータ入力補助機能が存在する。グラフデータ管理機能はグラフデータを管理する機能で、WebAPI を用いてグラフデータを管理できるため、様々なアプリケーションで API を用いることにより、グラフデータを管理できる。グラフデータ入力補助機能では、本システムを用いて学習者指導する指導者に対して、学習目標・学習項目の入力を容易に実施するための機能である。グラフデータ入力補助機能はフォームが木構造で入力することが可能で、学習目標・学習項目の入力が容易にできる。また、グラフデータ入力補助機能はグラフデータ管理機能の API を用いることにより、学習目標、学習項目をグラフデータへと変換し、グラフデータベースへとグラフデータを保存、および呼び出しを実行している。

1.1.5 評価実験の概要

1.1.6 本論文の構成

第 2 章 使用技術

2.1 使用技術

本章では、本研究で使用した技術について述べる。

2.1.1 Docker

概要

Docker[?] は、アプリケーションの開発、導入、実行するためのオープンなプラットフォームである。Docker を利用すれば、アプリケーションをインフラストラクチャーから分離できるため、ソフトウェアを素早く提供できる。また、Docker はアプリケーションをパッケージ化して実行するために、ほぼ分離された環境となるコンテナを提供している。

コンテナ

コンテナは Docker がアプリケーションを実行するための、ほぼ分離された環境を提供する。ほぼ分離された環境というものは namespace[?] で実現されている。

コンテナは namespace を用いることによって図 2.1 のように作成されている。

ホスト OS 上に namespaceA と namespaceB を作成し、それぞれの namespace の空間で全く独立したファイル構造やプロセス空間を持つことができる。このように分離された環境をコンテナという。コンテナのプロセス空間に関しては、実際にはホスト OS 上で動作しているプロセス ID とコンテナ上で動作しているプロセス ID をマッピングしているテーブルが存在する。これを用いてホスト OS 上では pid2 として認識されているが、コンテナ上では pid1 として認識されていることになる。

このことにより namespace を利用すると、プロセス空間やファイルシステムを一つの OS の中で分離できるので、namespace で区切られた部分であるコンテナは全く異なる OS のようにふるまうことが可能となる。

Docker Engine

2.1.1 節で述べたように Docker はコンテナを用いて動作している。しかし、プロセスをコンテナ上で稼働させるためには、OS の動作に必要なコマンドやライブラリが必要になる。これを実現するためには相当量のコマンドを発行しなければならない。そこで Docker は、Docker Engine を作成した。Docker Engine は主に以下のコンポーネントからなるクライアントサーバ型アプリケーションである。

- サーバ

長時間稼働する種類のプログラムでありデーモン・プロセスと呼ばれる。

- REST API

プログラムとデーモンとの間での通信方法を定義し、何をなすべきかを指示する。

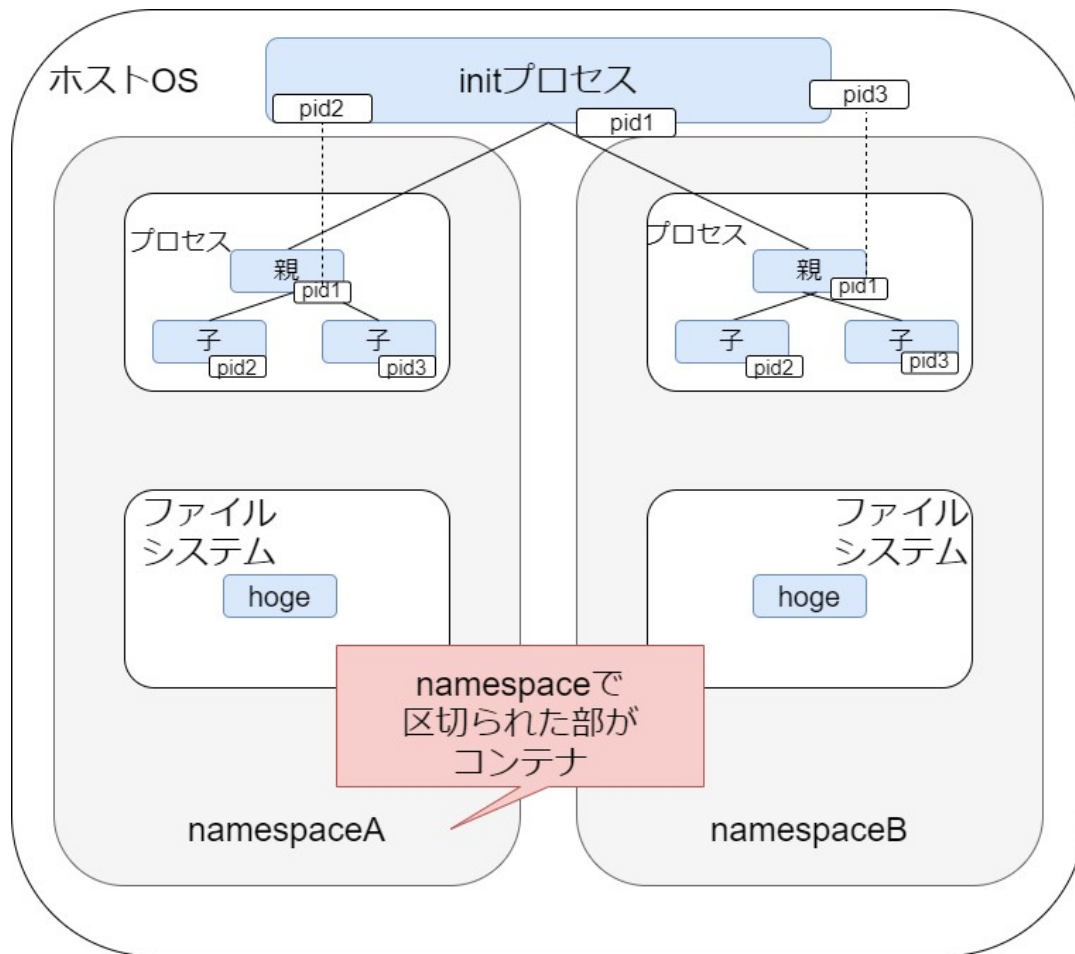


図 2.1 コンテナの構成図

・コマンドライン・インターフェイス (CLI)

クライアント: docker コマンドなど。

デーモンは、Docker オブジェクトを作成、管理する。Docker オブジェクトとは、イメージ、コンテナ、ネットワーク、データ、ボリュームなどを表す。

CLI は Docker REST API を通じて、スクリプトのコマンド実行により、Docker デーモンを制御、入出力を実行する。Docker アプリケーションの多くが、基本的なところでこの API や CLI を利用している。

また、Docker の内部には以下の 3 つのコンポーネントが存在する。

- Docker イメージ
- Docker レジストリ
- Docker コンテナ

Docker イメージとは、読み込み専用なテンプレートである。Docker イメージは Docker コンテナ作成時に使用される。各イメージはレイヤの積み重ねで構成される。また、新しいイメージの構築や既存のイメージの更新だけでなく、他人が作成した Docker イメージをダウンロードして使用することも可能である。Docker は UnionFS[?] を用いて各イメージのレイヤを単一のイメージに連結する。これにより、Docker イメージに変更を加えた場合、変更したい部分のレイヤを追加もしくは更新するだけでよいため、構築に時間がかからず早く簡単にイメージを提供できる。

Docker レジストリとは、イメージを保持するものである。パブリックもしくはプライベートに保管されているイメージのアップデートやダウンロードを実行することができる。パブリックな Docker レジストリとして Docker Hub[?] が提供されている。

Docker コンテナとは、ディレクトリと似たようなもので、アプリケーションの実行に必要なすべてが含まれている。各コンテナは Docker イメージによって作成される。Docker コンテナは実行・開始・停止・移動・削除できる。各コンテナは隔離されているため、安全なアプリケーションのプラットフォームとして動作する。

Docker Compose

Compose とは、複数のコンテナを定義し実行する Docker アプリケーションの為のツールである。Compose においては YAML[?] ファイルを使ってアプリケーションサービスの設定する。コマンド一つ実行するだけで、設定内容に基づいたアプリケーションを生成、起動する。

Compose を使うには基本的に 3 つのステップを踏む。

1. アプリケーション環境を Dockerfile に定義する。
2. アプリケーションを構成するサービスを docker-compose.yml ファイル内に定義する。
3. docker-compose up を実行したら、Compose はアプリケーション全体を起動・実行する。

コマンド一覧

Docker と Docker Compose で使用する代表的なコマンド一覧とコマンドの説明を表 2.1 に示す。

2.1.2 Python

概要

Python[?] は Windows, Linux/Unix, Mac OS X などの主要なオペレーティングシステムおよび Java や .NET などの仮想環境でも動作するインタプリタ形式の、対話的な、オブジェクト指向プログラミング言語である。この言語には、モジュール、例外、動的な型付け、超高水準の動的データ型、およびクラスが取り入れられている。Python はオブジェクト指向プログラミングを超えて、手続き型プログラミングや関数型プログラミングなど複数のプログラミングパラダイムをサポートしている。また、多くのシステムコールやライブラリだけでなく、様々なウィンドウシステムへのインターフェースがあり、C[?] や C++[?] で拡張することもできる。

Django

Django[?] は、Python で実装されたオープンソースの Web アプリケーションフレームワークの一つである。Django が作られた時の目的として、複雑なデータベース主体のウェブサイトを簡単に構築するというものがある。これを実現するため、Django ではコンポーネントの再利用性、素早い開発の原則に力を入れている。Django には以下のような特徴がある。

- **高速な動作**

Django には標準で分散型のキャッシュシステムである memcached[?] が備え付けられており、キャッシュ機能が強力である。

- **フルスタック・フレームワーク**

Django には、Web アプリケーションの実装に必要な、ユーザ認証、管理画面、RSS フィードなどの機能があらかじめ含まれている。

- **セキュリティ的に安全な設計**

Django では、デフォルトでパスワードなどはハッシュ化しデータベースに格納する。また、SQL インジェクション、クロスサイトスクリプティング、クロスサイトリクエストフォージェリなどの多くの脆弱性についても保護を有効にしている。

- **自由に選択できるプラットフォーム**

Django は、そのすべてが Python から作成されている。これにより、Python が Linux, Windows, MacOS X などで実行できるように Django も多くのプラットフォームで動作可能である。

また、Django の機能構成として MVT(Model・View・Template) というものがある。Django は MVT のもと、ルーティングというリクエストを振り分ける機能を用いてアプリケーションを動作させている。Django がどのように動作するかを図 2.2 に示す。

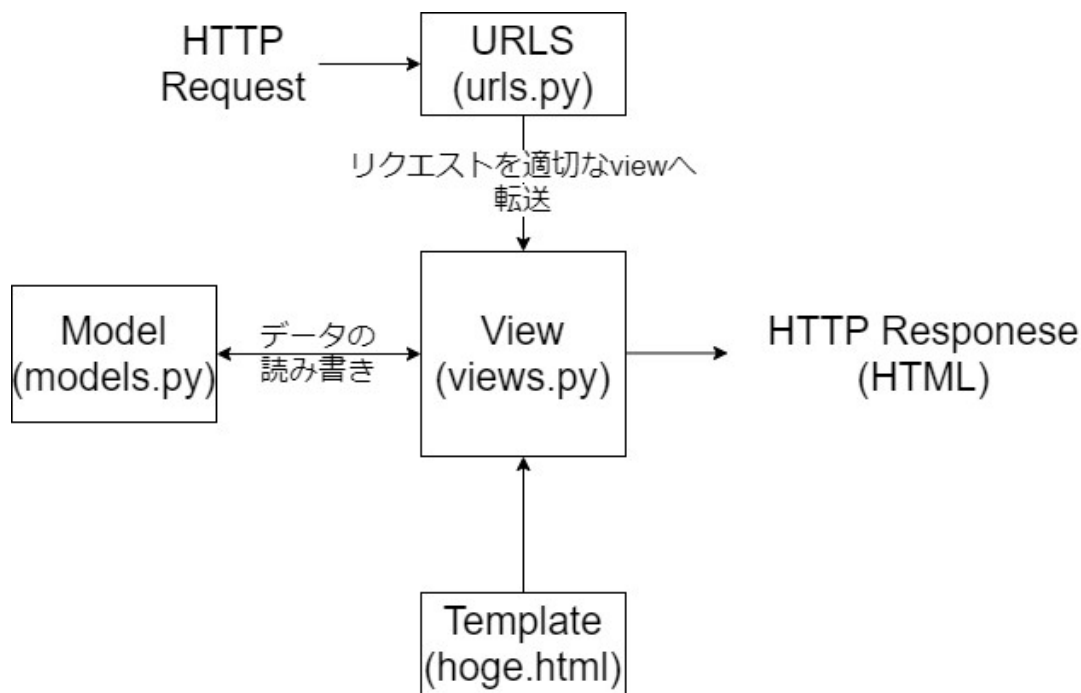


図 2.2 Django の動作の概要図

URLS は単一 URL から単一の関数を介してすべてのリクエストを処理することは可能だが、各リソースを処理する別々の View 関数を作成する方がメンテナンスが容易なため、URL をマッピングしている。これにより、URL に含まれる特定のパターンの文字列もしくは数字を照合し、これらをデータとして View 関数に渡して処理を行っている。

View は、HTTP リクエストを受け取り、HTTP レスポンスを返す関数である。View はモデルを介して要求を満たすために必要なデータにアクセスし、レスポンスのフォーマットを Template に委任する。

Model は、アプリケーションのデータ構造を定義し、データベース内のレコードを追加、変更、削除、および照会するための機能を提供する Python オブジェクトである。

Template は、ファイル構造やレイアウトを定義するテキストファイルで、プレースホルダを使用して実際のコンテンツを表示する。View は、モデルから取得したデータと HTML テンプレートを使用して動的に HTML ページを作成できる。また、Template を利用して、HTML だけではなく、あらゆる種類のファイルの構造を定義できる。

続いて、Django は複雑なデータベース主体のウェブサイトを簡単に構築するため、データベースを操作する 3 つの機能がある。

- Django シェル

Django アプリケーションの環境を有効にしたままコマンドで操作が可能。

- マイグレーション

データベースの操作を一括に実行・取り消すことが可能。

- 管理サイト

Web ブラウザで、データベースを操作することが可能。

これらの機能を用いることにより、複雑なデータベース主体のウェブサイトを簡単に構築できる。

最後に Django で主に使用するコマンド一覧とその説明を表 2.3 に示す。

Django REST framework

Django REST framework[?] とは Django を使って RESTful な API を開発するために利用されるライブラリである。Django REST framework を動作させるには、以下の要件を満たさなければならない。

1. Python (3.5, 3.6, 3.7, 3.8, 3.9)
2. Django (2.2, 3.0, 3.1)

2.1.3 その他関連技術

Go

Go[?] は静的型付けの効率と安全性、動的型付けのプログラミングの容易さを兼ね備えたプログラミング言語である。また、Go のコンパイラ、ツール、およびソースコードはすべてフリーなオープンソースである。

PostgreSQL

PostgreSQL[?] とは、Linux, Windows, MacOS X に対応した、オープンソースのリレーショナルデータベース管理システム (RDBMS) である。特徴として以下のようなものがある。

拡張性

PostgreSQL では、アーキテクチャが拡張可能である。このことにより、データベース内の手順のカスタマイズや自動化に使用できるユーザ定義関数とサードパーティライブラリへのサポートが可能となる。

同時実行性

PostgreSQL では、マルチバージョン同時実行制御があり、ユーザは同じデータベースの書

き込みと読み取りが可能となっている。

標準 SQL への準拠

PostgreSQL では、ISO/IEC の標準 SQL になった実装となっている。

Chart.js

Chart.js[?] とは、HTML5 の Canvas を使用したグラフ描画ライブラリである。Chart.js はオープンソースなプロジェクトで、8 つのチャートタイプと優れたレスポンスを有している。8 つのチャートタイプは以下の通りである。

- 線グラフ
- 棒グラフ
- レーダーチャート
- 円グラフ
- 鶏頭図
- バブルチャート
- 散布図
- エリアチャート

また、2 種類以上のチャートを複合した複合グラフも作成できる。

Json

Json[?](Javascript Object Notation) は、軽量なデータ交換フォーマットである。人間にとって読み書きが容易で、マシンにとってもパースや生成を行える。

NGINX

NGINX[?] は、フリーでオープンソースな Web サーバである。HTTP, HTTPS, SMTP, POP3, IMAP のリバースプロキシ機能や、ロードバランサ, HTTP キャッシュなどの機能を持つ。

Gunicorn

Gunicorn[?] は、UNIX のための Python WSGI HTTP Server である。WSGI[?] とは、Python において Web サーバと Web アプリケーションを接続するための、標準化されたインターフェース定義のことである。

表 2.1 Docker コマンド一覧

<code>docker run -it [イメージ名] /bin/bash</code>	docker コンテナを起動し、ログイン
<code>docker run -p [外部ポート]:[コンテナポート]</code>	コンテナに外部からアクセスする
<code>docker run -v [ホストボリュームパス]:[コンテナ内パス] [コンテナイメージ名]</code>	コンテナのボリュームをホストにマウントする
<code>docker save [イメージ名] >[ファイル名].tar.gz</code>	docker イメージを tar.gz 形式で保存する
<code>docker load <[ファイル名].tar.gz</code>	tar.gz 形式の docker イメージを取り込む
<code>docker exec -it [コンテナ ID/コンテナ名] /bin/bash</code>	起動中のコンテナにログイン
<code>docker cp [ホストファイルパス] [コンテナ ID]:[コンテナコピー先パス]</code>	起動中コンテナにホストからファイルをコピー
<code>docker commit [起動中のコンテナ ID] [イメージ名]</code>	起動中のコンテナをイメージとして保存
<code>docker start [コンテナ名]</code>	コンテナの開始
<code>docker stop [コンテナ名]</code>	コンテナの停止
<code>docker restart [コンテナ名]</code>	コンテナの再起動

表 2.2 Docker Compose コマンド一覧

docker-compose ps	コンテナの一覧を表示
docker-compose images	イメージの一覧を表示
docker-compose up --build	コンテナのビルドと起動
docker-compose down	docker-compose.yml で起動したコンテナの削除
docker-compose start	docker-compose.yml でビルドしたコンテナの起動
docker-compse stop [サービス名]	docker-compose.yml で起動しているコンテナの停止
docker-compse exec [サービス名] [コマンド]	docker-compose.yml で起動したコンテナにログインしてコマンドを発行

表 2.3 Docker コマンド一覧

manage.py createsuperuser	管理者ユーザを作成.
manage.py changepassword	ユーザのパスワードを変更.
manage.py check	プロジェクトの構成チェック.
manage.py dbshell	データベース管理用のシェルを起動.
manage.py dumpdata	データベースの内容を Json 等の形式でエクスポート.
manage.py loaddata	データベースに dump ファイルをインポート.
manage.py makemigrations	マイグレーションファイルの作成.
manage.py migrate	データベースにマイグレーションを適用.
manage.py shell	管理者用のコマンドシェルを起動し対話モードに入る.
manage.py startapp	新規アプリケーションの作成.
manage.py startproject	プロジェクトディレクトリの作成.
manage.py test	テストフレームワークの実行.
manage.py testserver	検証用サーバを起動.
manage.py collectstatic	static ファイルの収集.
manage.py runserver	アプリケーションの起動.

2.2 研究内容

本章では、概要、開発環境、関連研究、システム概要を述べた後に、開発した本プラットフォームの詳細について述べる。

2.2.1 概要

本研究の目的は、情報倫理教育に関する学習を支援することである。そのため、学習者が情報倫理に関して学ぶとき、手軽に学習する環境を構築する必要がある。これを解決するために本研究では学習をインターネット上で手軽に学習するために web アプリケーションを用いた e ラーニングプラットフォームを開発した。本プラットフォームは、情報倫理に関して学習する学習者と情報倫理に関するコンテンツを提供する教材提供者を対象としたシステムである。教材提供者に対して本プラットフォームではコンテンツ提供機能、統計情報提供機能、コンテナ管理機能を用意している。教材提供者はこれらの機能を用いることにより情報倫理に関するコンテンツを web アプリケーション上に投稿でき、学習者はそれらのコンテンツを用いて情報倫理に関する学習を行える。

2.2.2 開発環境

本プラットフォームを作成するにあたって使用した PC のスペックと開発環境を表 2.4 に示す。

表 2.4 PC のスペックと開発環境

CPU	Intel Core i7 @ 3.70GHz
Memory	24.0GB
OS	Windows 10 Education 64-bit
開発環境	Docker version 20.10.0, build 7287ab3 docker-compose version 1.27.4, build 40524192

クライアントシステムとサーバシステムの開発環境

クライアントシステムとサーバシステムの開発には Docker のコンテナを用いた。表 2.5、表 2.6 にそれぞれの開発環境を示す。

表 2.5 クライアントシステムの開発環境

OS	Alpine Linux(3.12.3)
使用言語・フレームワーク	Python 3.7.9 Django 3.0.2

表 2.6 サーバシステムの開発環境

OS	Debian GNU/Linux 10 (buster)
使用言語・フレームワーク	go version go1.15.6 linux/amd64 djangorestframework 3.12.1

2.2.3 関連研究

関連研究として、上田氏らの「倫倫姫プロジェクト-学人連携 Moodle による多言語情報倫理 e ラーニング-」[?] がある。

倫倫姫プロジェクトでは大学の情報倫理教育における以下 3 つの問題を解決している。

- (i) 標準化と可視化がなされていない
- (ii) 留学生への教育が困難
- (iii) 持続可能性が低い

(i) を解決するために、倫倫姫ではサンプル規程集「A3301 教育テキスト作成ガイドライン (一般利用者向け)」に準拠することで、内容を標準化した。また、受講履歴を閲覧でき受講者の学習状況の可視化も実現した。

(ii) を解決するために、倫倫姫では英語、中国語、韓国版を作成しており、各言語圏の文化の違いも考慮しコンテンツを作成することで解決した。

(iii) を解決するために、倫倫姫では SCORM が規定する e ラーニングコンテンツパッケージを利用した。これは、パッケージ構造とリソースを記述するマニフェストファイル (imsmanifest.xml) とそれから参照される物理ファイル (HTML, swf など) をファイル単位で修正可能なため継続的な改訂を可能としている。

本研究でも (i), (iii) については同様に本プラットフォームで解決している。(i) に関しては、コンテンツ提供機能により、教材提供者間でコンテンツを共有できることから解決している。(iii) に関しては、本プラットフォームではコンテンツは Web アプリケーション上で管理しており、コンテンツの修正も Web 上から行えることから解決している。

一方、本プラットフォームにおいては、コンテンツに関してはマークダウン形式で記入可能なことから、過去のコンテンツを本プラットフォーム上に移行することが簡単である。また、コンテンツ管理機能により、過去に作成したアプリケーションに関しても本プラットフォーム上に移行することが簡単である。

2.2.4 システム概要

要件定義

本システムはサーバ、クライアントで以下の要件を満たす必要がある。
まずサーバの要件を示す。

- Docker の動作: すべての開発に Docker を用いる
- Web サーバの動作: Web アプリケーションの公開に用いる
- Python の動作: Django の動作に用いる
- Go の動作: API の動作に用いる
- PostgreSQL の動作: ユーザの管理等のデータベースとして用いる

サーバは Web サーバを用いてクライアントをネットワークに公開し、Docker を用いて Python と Go と PostgreSQL を動作させる。Python は Django を動作させ、クライアントに GUI 等を提供する。Go は API 作成に用いて、Docker を操作する。PostgreSQL は、Django を通してクライアントから入力されたデータ等を保存する。

続いて、クライアントの要件を示す。

- html5 が解釈できる Web ブラウザの動作: クライアントの動作環境となる

クライアントは Web アプリケーションであり、html5 を解釈できる Web ブラウザ上で動作する。クライアントは教材提供者と学習者に対して GUI を提供する。詳細は、2.2.5 節、2.2.6 節、2.2.7 節で述べる。

また、作成する機能は以下の要件を満たすように作成した。

- 日々変化する情報倫理に対応する為の機能を作成する
- 年代、性別の違いによる情報倫理に関する意識の違いを確認する為の機能を作成する
- 本アプリケーション以外のアプリケーションを本アプリケーションでも利用可能にする機能を作成する

日々変化する情報倫理に対応する為の機能は、コンテンツ提供機能として作成し、これを用いてコンテンツを素早く投稿、編集する。年代、性別の違いによる情報倫理に関する意識の違いを確認する為の機能は、統計情報提供機能として作成し、年代や性別をグラフとして提示する。本アプリケーション以外のアプリケーションを本アプリケーションでも利用可能にする機能は、コンテンツ管理機能として作成し、Docker を用いたアプリケーションを本プラットフォーム上でも利用可能にする。

システム構成

本システムの構成を図 2.3 に示す。本システムは Django で構成されるアプリケーション部 (以下, クライアント), および Go と djangoestframework で構成されるサーバ部 (以下, サーバ) から構築される。サーバは API を用いて統計情報提供機能とコンテナ管理機能を動作させる。

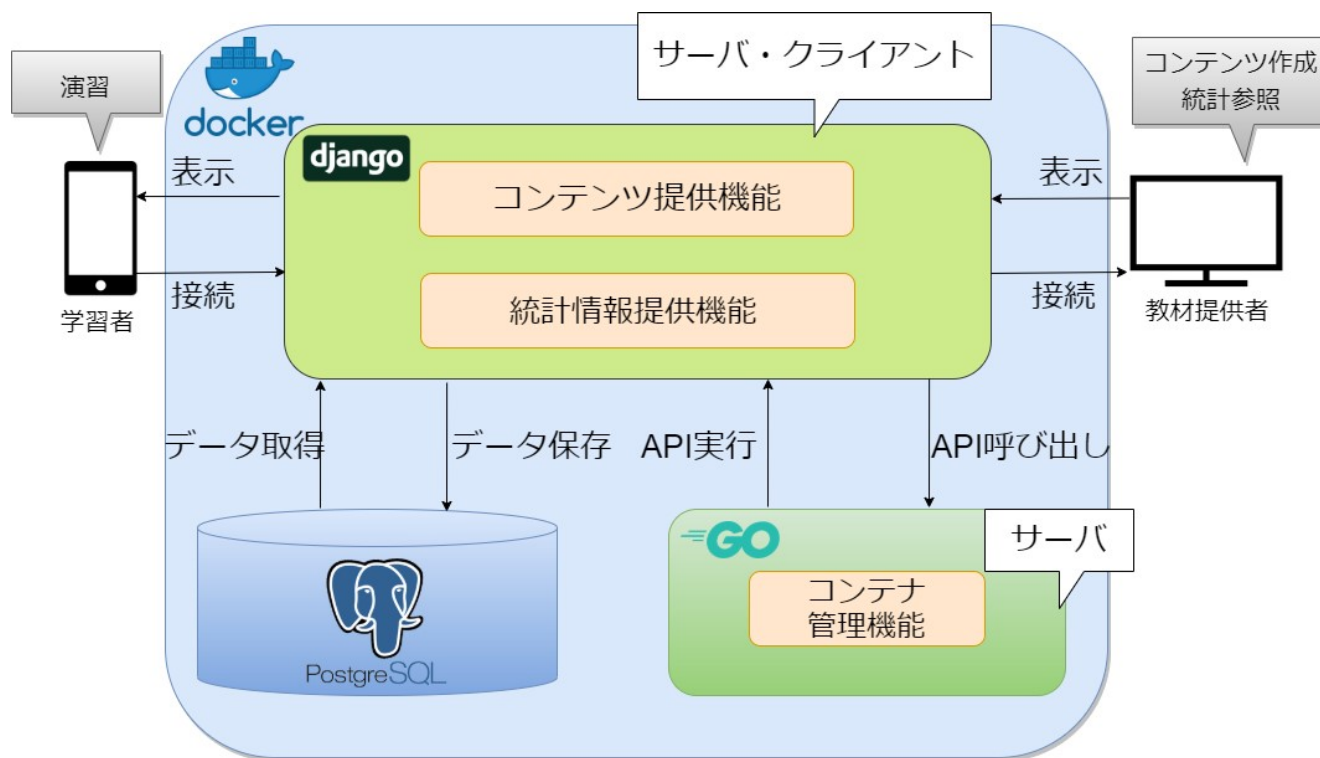


図 2.3 システム構成

コンテンツ提供機能の GUI を図 2.4 に, 統計情報提供機能の GUI を図 2.5 に, コンテナ管理機能の GUI を図 2.6 にそれぞれ示す。

教材提供者は図 2.4 のコンテンツ提供機能を用いて情報倫理に関するコンテンツを提供する。

admin! ユーザー情報更新 パスワードの変更 ログアウト コンテンツ関連 ▾ 統計情報

タイトル:

ネット詐欺などに巻き込まれないようにするために

タグ:

ネット依存
コンテンツ依存
つながり依存
ネット詐欺

説明文:

ネット詐欺

高校1年生のゆきさんとめぐ みさんは、インターネットでアイドルグループのコンサート チケットを購入しましたが、行ってみるとそれが偽物だったことが判明しました。母親のクレジットカードには、50 万円 もの決済書が送られてきました。

<iframe width="560" height="315" src="https://www.youtube.com/embed/0P26mr9zkVs" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

ネット詐欺

高校1年生のゆきさんとめぐ みさんは、インターネットでアイドルグループのコンサート チケットを購入しましたが、行ってみるとそれが偽物だったことが判明しました。母親のクレジットカードには、50 万円 もの決済書が送られてきました。

後で見る 共有

図 2.4 コンテンツ提供機能の GUI

図 2.5 の統計情報提供機能では，教材提供者が学習者の回答情報等をグラフとして確認できる．



図 2.5 統計情報提供機能の GUI

図 2.6 のコンテナ管理機能では教材提供者が Docker を用いて作成した他の教育アプリケーションを本プラットフォームでも利用することが可能となる。

admin! ユーザー情報更新 パスワードの変更 ログアウト コンテンツ関連 ▼ 統計情報

URL:

使用可能ポート一覧

49152

49153

49154

49155

+

実行手順:

送信

図 2.6 コンテナ管理機能の GUI

学習者は図 2.4 のコンテンツ提供機能を用いて作成されたコンテンツを図 2.7 のようにして閲覧、学習することが可能である。



図 2.7 コンテンツ閲覧時の GUI

システムの内部構成

クライアントの内部構成を図 2.8 に示す。はじめに、教材提供者および学習者が各々ネットワークに接続できる環境を用意し、Web ブラウザを立ち上げ、特定の IP アドレスを入力しログインする。ただし、教材提供者が本プラットフォームの機能を利用するにはログインは必須であるが、学習者は必須ではない。

Web ブラウザで本プラットフォームに接続しログインした後、教材提供者は GUI のコンテンツ提供機能、統計情報提供機能、コンテナ管理機能を使用できる。コンテンツ提供機能は教材提供者が入力した内容をサーバを介しデータベースに保存する。統計情報提供機能はデータベースからサーバを介しデータを取得し Web ブラウザ上に情報を表示する。コンテナ管理機能は教材提供者が入力した内容を Go で作成した API で実行し、Docker のコマンドを用いてコンテナが建ち上がっていることを確認し、建ち上がっていた場合それにアクセスする URL を画面上に発行する。



図 2.8 クライアントの内部構成

続いて、サーバの内部構成を図 2.9 に示す。サーバは Django を用いて作成された Django 処理部と Go を用いて作成された Go 処理部、データを保存するためのデータベース処理部がある。サーバはクライアントからの通信が行われた場合に動作する。それぞれの処理部の内容について以下に示す。

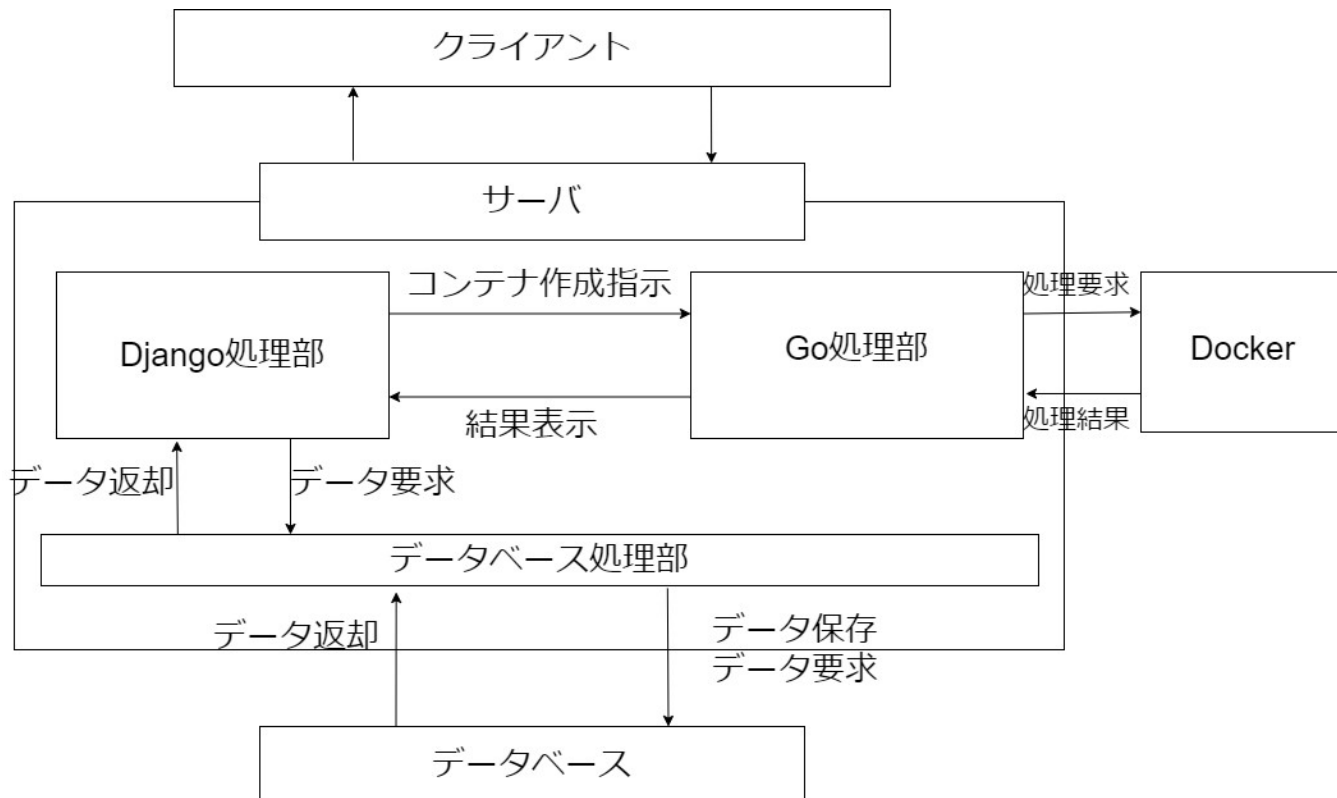


図 2.9 サーバの内部構成

- Django 処理部

Django 処理部では、クライアントからフォームに従って入力されたデータをデータベースに登録、抽出する処理を行っている。具体的には、ユーザの新規登録、ログイン処理、パスワードや登録情報の変更、コンテンツ提供機能とコンテナ管理機能で入力された情報の登録と修正、タグの検索処理、統計情報提供機能のためのデータの検索が挙げられる。

- Go 処理部

Go 処理部では、コンテナ管理機能により入力された情報を API を用いて Docker に実行、処理させコンテナを建ち上げている。

- データベース処理部

データベース処理部では、Django 処理部においてデータベースのやり取りが必要な場合に動作する。Django 処理部から sql コマンドが発行され、それを実行処理している。

システムのフローチャート

教材提供者のフローチャートを図 2.10 に，学習者のフローチャートを図 2.11 に示す．本システムはまず，教材提供者はログインが必須，学習者は任意となっている．そのため今回は簡単のため教材提供者と学習者がともにログインした場合のフローチャートとする．

はじめに教材提供者について説明する．教材提供者は事前にユーザ登録を行ったことを管理者に通知し，管理者からアカウントを一般のものから教材提供者のアカウントに設定する必要がある．教材提供者アカウントに設定した後，ログイン処理を実行する．ログインが完了したら教材提供者はコンテンツ提供機能を用いてコンテンツの必要情報を入力し，サーバに登録する．統計情報提供機能を用いる場合は，コンテンツに対する統計情報をサーバに対しリクエストすると，その結果が画面上に表示される．コンテンツ管理機能を用いる場合は，動作させたいアプリケーションの情報をクライアント上で入力することにより，サーバに対しその情報がリクエストされ，実行される．実行が正常に完了した場合，接続するための URL が画面上に表示される．

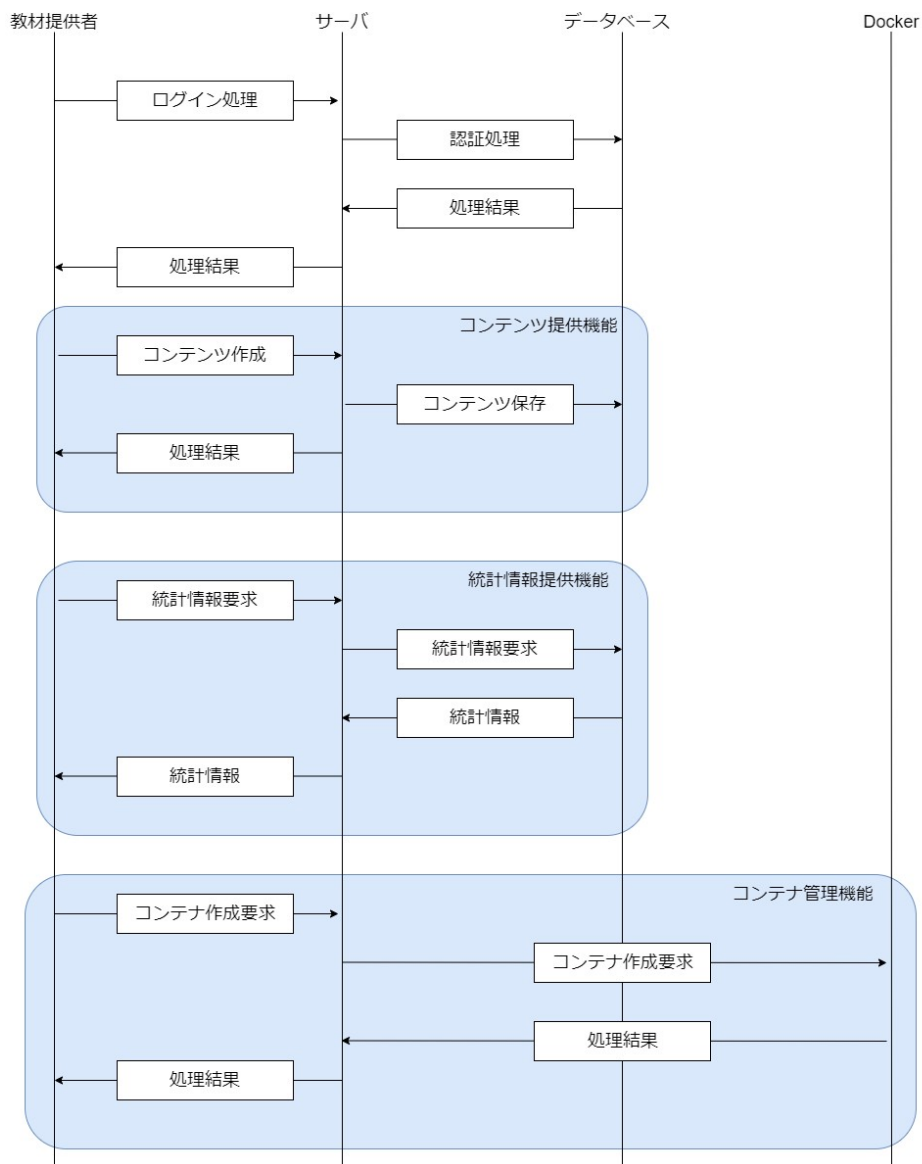


図 2.10 教材提供者のフローチャート

続いて学習者について説明する。学習者は Web ブラウザで本プラットフォームに接続した後、本プラットフォームでアカウント作成できる。アカウント作成には、本プラットフォーム上で表示されるユーザ名、性別、年齢、パスワードと確認用パスワードが必要である。アカウント作成後、トップページにて教材提供者が作成したコンテンツを選択、閲覧できる。

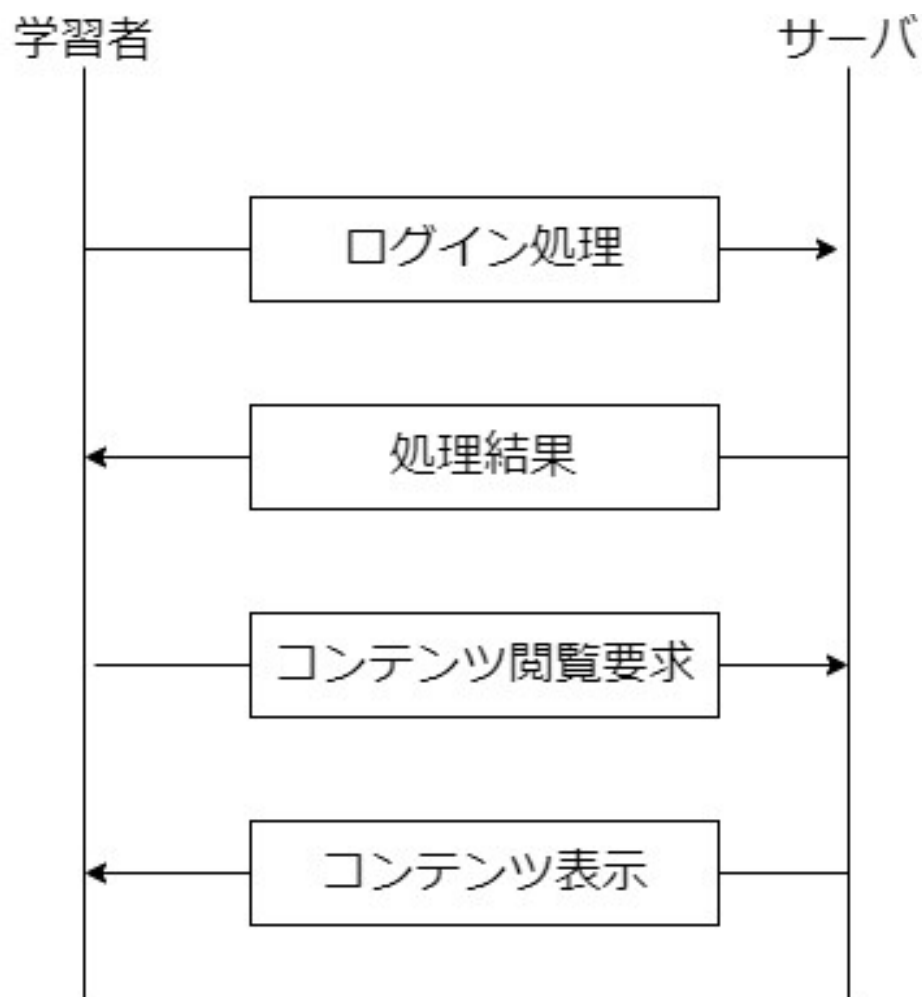


図 2.11 学習者のフローチャート

2.2.5 コンテンツ提供機能

日々変化する情報倫理に対応するために、コンテンツを素早く投稿、編集する必要がある。それに対応するため、教材提供者が情報倫理に関するコンテンツを Web アプリケーション上で提供するためにコンテンツ提供機能を作成した。本機能では Web 上で教材提供者のみがコンテンツの投稿と管理ができる。コンテンツを投稿するときの GUI を図 2.12 に示す。図 2.12 に示したとおり、コンテンツを投稿するときに必要な情報は、コンテンツのタイトルとコンテンツのタグ、本文である。本文はマークダウン形式で記入可能であり、画像や動画の挿入が容易にできる。またプレビュー機能もあるため投稿する前にどのような見で投稿されるのかを確認できる。

タイトル:

このサイトについて

タグ:

ネットゲーム
ゲーム依存
ネット依存
コンテンツ依存

+

説明文:

序論

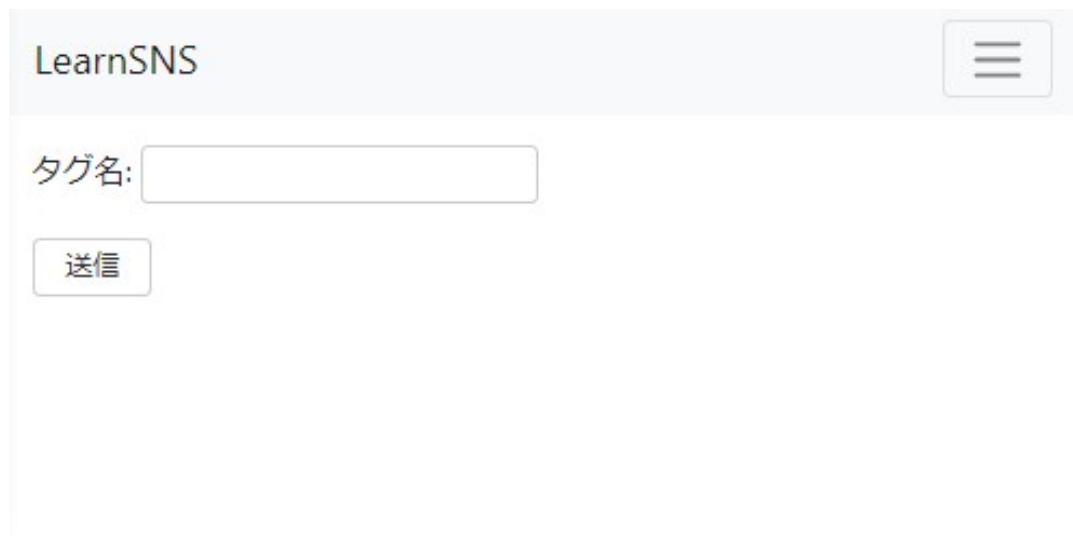
総務省が令和元年に 37182 人に行った調査によると、インターネットの利用率は、その約 9 割にまで増加している 1)。その理由の 1 つに SNS の普及が挙げられる。SNSにおいて平成27年にみずほ情報総研株式会社が 1178 人に行った調査研究 2)によると、SNS 上でトラブルの経験があると回答した割合は 15%程であった。トラブルの内容は、自分自身の発言が他人に異なる意味で受け取られてしまう、自分の意志とは関係なく個人情報などが他人に公開されてしまうなどである。このようなトラブルを避けるために情報倫理教育は有効である。情報倫理教育の内容として個人情報の保護、人権侵害、コンピュータ犯罪などがある。これらの教育は特にインターネットの利用において重要である 3)。そこで本研究では、情報倫理に関する学習を支援することを目的に、情報倫理に関するコンテンツを提供可能とするプラットフォーム(以下、本プラットフォーム)を開発した。本プラットフォームを用いることで、情報倫理に関するコンテンツを web 上で管理、提供できる。これにより情報倫理を学ぶ際、コンテンツを用いて学習することにより、トラブルの減少やリテラシーの向上が期待できる。

序論

総務省が令和元年に 37182 人に行った調査によると、インターネットの利用率は、その約 9 割にまで増加している 1)。その理由の 1 つに SNS の普及が挙げられる。SNSにおいて平成27年にみずほ情報総研株式会社が 1178 人に行った調査研究 2)によると、SNS 上でトラブルの経験があると回答した割合は 15%程であった。トラブルの内容は、自分自身の発言が他人に異なる意味で受け取られてしまう、自分の意志とは関係なく個人情報などが他人に公開されてしまうなどである。このようなトラブルを避けるために情報倫理教育は有効である。情報倫理教育の内容として個人情報の保護、人権侵害、コンピュータ犯罪などがある。これらの教育は特にインターネットの利用において重要である 3)。そこで本研究では、情報倫理に関する学習を支援することを目的に、情報倫理に関するコンテンツを提供可能とするプラットフォーム(以下、本プラットフォーム)を開発した。本プラットフォームを用いることで、情報倫理に関するコンテンツを web 上で管理、提供できる。これにより情報倫理を学ぶ際、コンテンツを用いて学習することにより、トラブルの減少やリテラシーの向上が期待できる。

図 2.12 コンテンツ提供機能の GUI

コンテンツのタグは図 2.12 に示すプラスボタンを押下することにより、別画面で新たなタグを登録できる。タグを登録するときの GUI を図 2.13 に示す。



LearnSNS

タグ名:

送信

図 2.13 タグ登録の GUI

また，教材提供者はコンテンツを管理するためのグループに参加する．そのグループはすべてのコンテンツに対して公開または非公開を選択することでコンテンツを管理する．グループに参加している教材提供者全員が認可したコンテンツのみが学習者に公開される．これにより，コンテンツの正当性を担保できる．教材提供者をグループに参加させるには，図 2.14 のようにして教材提供者のアカウントを選択し教材提供者のグループに登録する．

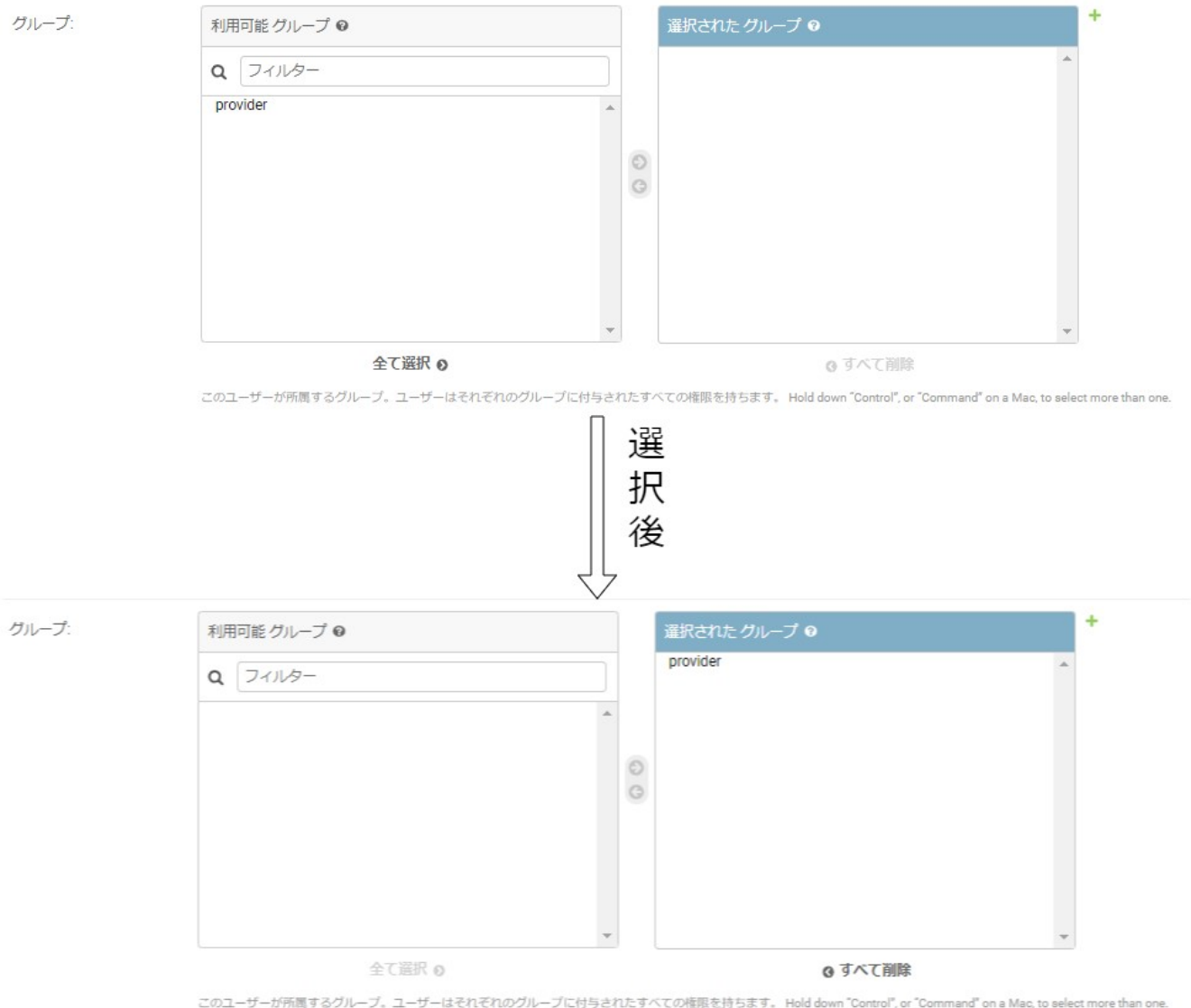


図 2.14 グループ登録の GUI

続いて本機能ではコンテンツ作成後にコンテンツの 4 択問題を作成することができる。問題の作成には、問題のタイトル、問題文、選択肢 1~4 および正解の選択肢をフォームに従って入力する。問題作成時の GUI を図 2.15 に示す。

最後に、Django の機能を用いることにより本機能で投稿したコンテンツを Json ファイルに変換することができる。Json ファイルに変換するには、以下のような手順を経てコマンドを発行すればよい。これにより、教材提供者はコンテンツを互いに Json ファイルを介して共有することが可能となる。

1. 「docker exec -it コンテナ名 /bin/ash」などとして、本プラットフォームを動作させている

LearnSNS ようこそ、admin! ユーザー情報更新 パスワードの変更 ログアウト コンテンツ関連 ▾ 統計情報

問題のタイトル:

問題文:

選択肢1:

選択肢2:

選択肢3:

選択肢4:

答え:

▼

登録

図 2.15 問題作成の GUI

コンテナにログインする。

2. cd コマンドを用いて Django によって自動的に生成された manage.py ファイルと同階層に移動する。
3. 「python manage.py dumpdata Json ファイルに変換したいデータの DB のテーブル名 > 保存する Json ファイル名.json」を実行後、Json ファイルが生成される。

2.2.6 統計情報提供機能

統計情報提供機能は、年代、性別の違いから情報倫理に関する意識の違いが生まれていることから [?], 教材提供者が作成した問題を学習者が解いたときの回答情報を基に、グラフで統計情報を提供する機能である。これにより、教材提供者は作成したコンテンツの質を向上させることができる。提示する統計情報の内容としては、問題の各選択肢における割合や回答者の年齢層、性別である。統計情報提供機能の GUI は図 2.16 の通りである。



図 2.16 統計情報提供機能の GUI

本機能は Chart.js という javascript で作成されたグラフ描画ライブラリを作成している。Chart.js では線グラフ、棒グラフ、レーダーチャート、鶏頭図、ドーナツチャート、円グラフ、バブルチャートを作成できる。本機能では、棒グラフと円グラフを使用している。

また、本機能で取得可能な情報を表 2.7 に示す。これらの取得できる情報を使って教材提供者は Chart.js で新たなグラフを挿入することができる。

表 2.7 取得可能情報一覧

取得可能情報	詳細
学習者情報	年齢
	性別
	回答した問題・選択肢
コンテンツ情報	問題に対応したコンテンツ

2.2.7 コンテナ管理機能

本プラットフォーム上で、教材提供者は過去のアプリケーションを用いたい場合がある。それに対応するため、教材提供者が Docker を用いて作成したアプリケーションを本プラットフォームでも利用可能にするためコンテナ管理機能を作成した。本機能を利用するには、本プラットフォームで動作させたいコンテナ情報が記載された Dockerfile と docker-compose.yml を事前に Github などのリポジトリに用意する。外部通信するためのポート番号は、本機能が提示するポート番号のみが利用可能となっている。以上の情報を、本機能が提示するフォームに入力することにより、本プラットフォームで Docker のコンテナを作成し、アクセスするための URL が画面上に表示される。教材提供者は発行された URL にアクセスすることで動作を確認できる。また、発行された URL をコンテンツ提供機能の本文に貼り付けることにより、学習者はそのコンテンツを利用できる。

本機能の入力画面と入力が完了し、コンテナが作成された後の画面を図 2.17、図 2.18 に示す。

The screenshot shows the 'Container Management' input form in the LearnSNS application. At the top, a navigation bar contains the text 'LearnSNS' and several links: 'ようこそ, admin!', 'ユーザー情報更新', 'パスワードの変更', 'ログアウト', 'コンテンツ関連', and '統計情報'. The form itself is divided into three sections: 1. 'URL:' with a text input field containing 'https://github.com/niye1123/netlab.git'. 2. 'ポート番号:' (Port Number) with a dropdown menu showing a list of ports: 49152, 49153, 49154, and 49155. 3. '実行手順:' (Execution Steps) with a text area containing the command 'docker-compose up -d --build'. A blue '送信' (Send) button is located at the bottom right of the form.

図 2.17 コンテナ管理機能の入力画面

LearnSNS ようこそ、admin! ユーザー情報更新 パスワードの変更 ログアウト コンテンツ関連 ▼ 統計情報

コンテナの作成が完了しました。

<http://localhost:49152>

図 2.18 コンテナ管理機能のコンテナ作成後の画面

2.3 実験・考察

本章では，実施した利用評価実験の内容と結果および考察について述べる．

2.3.1 利用評価実験

本プラットフォームが，学習者に対し情報倫理に関するコンテンツを提供でき，そのコンテンツを見て学ぶことにより情報倫理に関するトラブルの減少とリテラシーの向上が期待できるかどうかを確認するために，利用評価実験を行った．実験では20代男性10人に対して，本プラットフォームのコンテンツを事前に学習したコンテンツ利用者群と，事前に学習していないコンテンツ非利用者群の5人ずつに分かれて実験を行った．そして各々の群に対し，GoogleFormを用いて全5問の情報倫理に関する問題を解いてもらった．ただし，コンテンツ非利用者群に対してGoogleFormで問題を解くときに，わからない単語等があれば調べてもよいとした．

このような条件で実施したコンテンツ利用者群とコンテンツ非利用者群の実験結果を図2.19，図2.20に示す．図2.19はコンテンツ利用者群とコンテンツ非利用者群の獲得点数を表している．図2.20はコンテンツ利用者群とコンテンツ非利用者群の獲得点数の平均と標準偏差を表している．

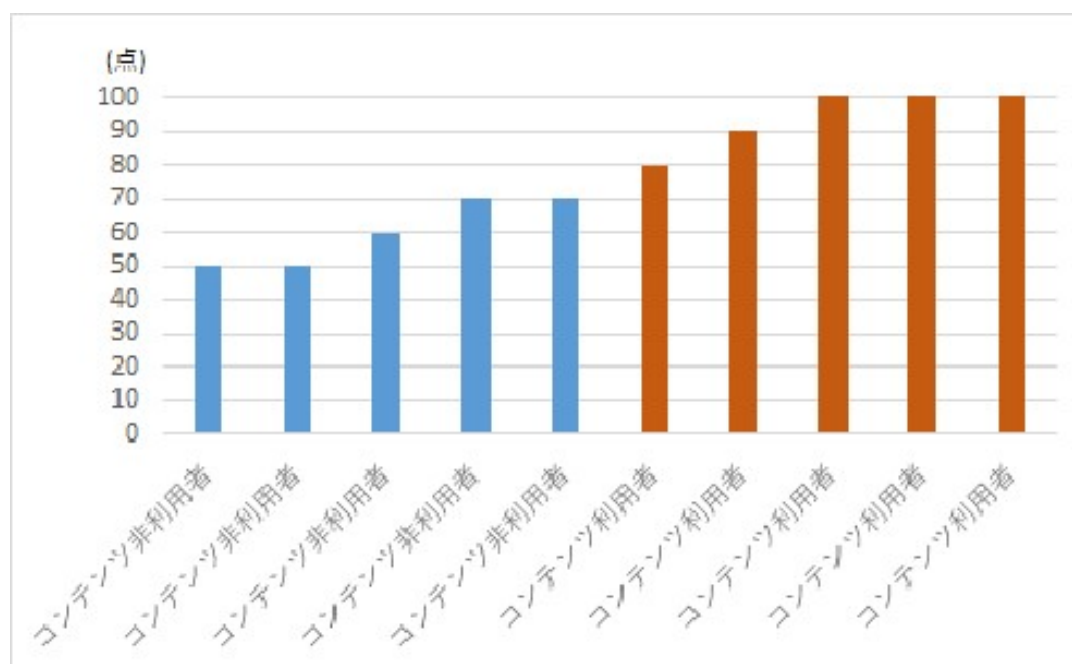


図 2.19 コンテンツ利用者群とコンテンツ非利用者群の獲得点数

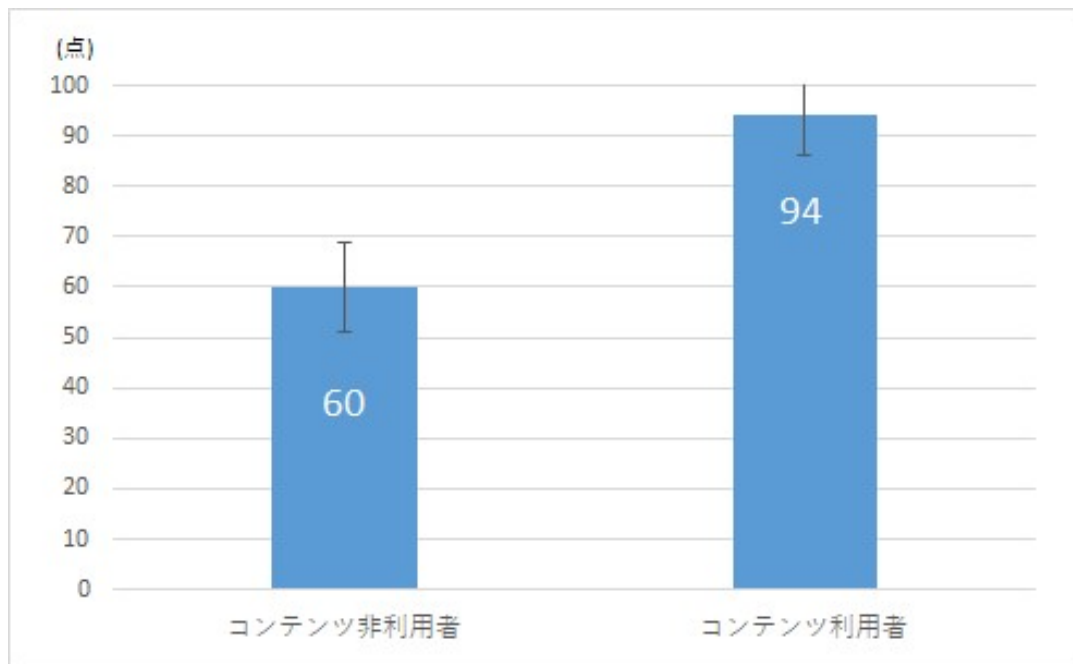


図 2.20 コンテンツ利用者群とコンテンツ非利用者群の獲得点数の平均と標準偏差

実験の結果、平均点においてコンテンツ利用者群が 94 点、コンテンツ非利用者群が 60 点と 34 点の差がついていることが分かった。また、標準偏差においてはコンテンツ利用者群が 8.00、コンテンツ非利用者群が 8.94 であり、0.94 の差がついている。以上から、本プラットフォームを用いて情報倫理について学習すると、点数のばらつきはコンテンツ利用者群の方がなく、点数も平均点の差からコンテンツ非利用者群と比べ高くなっていることがわかる。このことからコンテンツ非利用者群は、情報倫理に関して単語の記憶、理解だけを行っておりその結果情報倫理に対して十分な理解を得られずに問題を解き点数が低くなったと考えられる。一方、コンテンツ利用者群は、本プラットフォームで提示したコンテンツを学んでいる。このコンテンツには実際に起こりうる情報倫理に関する問題をシナリオ形式で提示している。このことによりコンテンツ利用者群は情報倫理に関して深く理解を得られたのではないかと考えられる。したがって、本プラットフォームは学習者に対して、情報倫理に関するトラブルの減少とリテラシー向上を期待できる。

2.4 結論・今後の課題

本研究では、持続的な学習環境を提供することを目的に、情報倫理教育における e ラーニングのためのプラットフォームを開発する。まず、本稿ではコンテンツの作成、統計情報の確認、外部アプリケーションの導入を補佐する機能を開発した。本プラットフォームを用いることで、情報倫理に関するコンテンツを web 上で管理、提供でき、持続的にコンテンツの提供ができる。これらのコンテンツを用いて学習することにより、トラブルの減少やリテラシーの理解を期待できる。

今後の予定として、教育効果を高めると言われるアニメーションを利用し [?], 本プラットフォーム上で情報倫理に関するアニメーション作成の支援が行えるような機能を作成する予定である。

謝辞

本研究を遂行するに当たり，熱心な御指導および御鞭撻をいただきました井口信和教授に深く感謝いたします。また，実験にご協力いただきました被験者の方々に深く感謝いたします。さらに，ネットワーク研究室の皆様をはじめ，研究活動を支えてくださりました皆様に深く感謝いたします。

参考文献

- [1] 文部科学省. 教育の情報化に関する手引」について. 入手先<https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/mext_00117.html>. 参照 (2022-01-15).
- [2] 三浦邦夫. e ラーニングの導入方法. 経団連, 5 1995.
- [3] 油谷知岐, 瀬田和久, 林佑樹, 池田満. セマンティックな教材による学習目標設計スキル育成支援. JSiSE2020 第 45 回全国大会, pp. 65–66, 2020.
- [4] J. D. Novak and D. B. Gowin. *Learning How to Learn*. Cambridge University, 1984.
- [5] 東本崇仁, 今井功, 堀口知也, 平崎宗. 誤りの可視化による階層構造の理解を指向したコンセプトマップ構築学習の支援環境. 教育システム情報学会誌, Vol. 30, No. 1, pp. 42–53, 2013.
- [6] 野村湧司, 林利憲, 荻野哲男, 角谷和俊. 教育コンテンツ上での理解度の再確認支援方法. 情報処理学会研究報告, Vol. 2017, No. 5, pp. 1–6, 2017.
- [7] 平塚紘一郎, 田中洋一, 澤崎敏文. 「学習成果の可視化」システムの構築— e ポートフォリオ mahara の活用 —. 仁愛女子短期大学研究紀要, Vol. 45, pp. 25–29, 2012.
- [8] Tsukasa Hirashima, Kazuya Yamasaki, Hiroyuki Fukuda, and Hideo Funaoi. amework of kit-build concept map for automatic diagnosis and its preliminary us. *Research and Practice in Technology Enhanced Learning, APSCE*, Vol. 10, No. 1, pp. 1–21, 2015.
- [9] 西川孝二, 塩谷将一, 下川尚亮, 岡田信人, 石田崇. グラフデータベース neo4j による学習ポートフォリオ作成支援システム. 計測自動制御学会 システム・情報部門 学術講演会 2015 講演論文集, Vol. SS22-3, pp. 486–487, 2015.