

令和 3 年度 卒業論文

認知行動療法 3 コマ漫画
認知行動療法学習アプリの制作

近畿大学工学部 情報学科

情報物理研究室

1810990019 戸田正彦

目次

- 1 はじめに
 - 1.1 研究のテーマ
 - 1.2 研究の背景
 - 1.3 研究の目的
 - 1.4 高次機能障害
 - 1.5 認知行動療法
 - 1.5.1 認知療法
 - 1.5.2 論理療法
- 2 使用技術
 - 2.1 使用言語
 - 2.1.1 JavaScript
 - 2.1.2 Node.js
 - 2.1.3 PHP
 - 2.2 ライブラリ・フレームワーク
 - 2.2.1 Vue.js
 - 2.2.2 VueCLI
 - 2.2.3 Firebase
 - 2.2.4 Flicking
 - 2.2.5 Bootstrap Icons
 - 2.3 使用技術
 - 2.3.1 SPA
 - 2.3.2 PWA
 - 2.4 使用ツール・サービス
 - 2.4.1 Git・GitHub
- 3 研究内容
 - 3.1 アプリケーションの概要
 - 3.1.1 アプリのフローチャート
 - 3.1.2 ユーザーと管理者
 - 3.1.3 漫画コンテンツ
 - 3.2 開発環境の構成
 - 3.2.1 ディレクトリ構成
 - 3.2.2 ルーティング
 - 3.2.3 状態管理
 - 3.3 コンポーネント
 - 3.3.1 ヘッダー
 - 3.3.2 フッター
 - 3.4 サイト構成
 - 3.4.1 ログイン画面
 - 3.4.2 ホーム
 - 3.4.3 サイト情報
 - 3.4.4 漫画コンテンツ
 - 3.4.4.1 Flicking
 - 3.4.5 ユーザー設定
 - 3.4.6 グループ情報
 - 3.4.6.1 グループ情報
 - 3.4.6.2 メンバーリスト
 - 3.4.6.3 グループの編集
 - 3.4.7 ログ

3.4.7.1 進行度表

3.4.7.2 履歴

4 まとめ

4.1 改善点

5 謝辞

6 参考文献

1. はじめに

1.1. 研究のテーマ : 「認知行動療法 3 コマ漫画アプリ」

本研究では、認知行動療法について研究し、認知行動療法を適用した Web アプリケーション「認知行動療法 3 コマ漫画」を開発する。認知行動療法は、感情が発生するまでの一つの**出来事**における物の見方（**認知**）と**感情**の関係に着目し、**認知の歪み**を治すことによって、偏った感情や行動を変化させる治療方法である。3 コマ漫画コンテンツを出来事・認知・感情の3つのセクションに分け、3 コマの漫画にすることで、認知行動療法の効果を高めることを目指す。

1.2. 研究の背景

情報物理研究室では月に一度、心理学の教授と広島県立リハビリテーションセンターの職員の方々と心理学ミーティングを行っている。このミーティングでは、高次機能障害における認知行動療法の適用について議論が行われている。その議論の中で案として提案された、認知行動療法を適用した3コマ漫画の Web アプリケーションを開発することにした。このアプリケーションは、高次機能障害を有する患者だけでなく、すべての人が使えるようにしている。その理由として、認知行動療法について理解してもらうことで、ストレスの軽減や自身の認知に対する理解ができると考えたためである。

1.3. 研究の目的

本研究で開発した Web アプリケーション「認知行動療法 3 コマ漫画」の目的は、**感情は認知に左右される**ことを学習することで、自身の認知の癖について認識してもらうことである。

本システムでは、認知行動療法の一種である認知療法に焦点をあて、認知行動療法を行える Web アプリケーションの開発を行う。本システムの対象者は、一般の人の他に、悲観的な考え方や心理状態の患者も含んでいる。その例として高次機能障害やうつ病を有する患者が挙げられる。そのため、高次機能障害を有する患者に配慮したユーザーインタフェースを搭載している。例として、視覚の左右どちらか半分しか認識できない半側空間無視を考慮したユーザーインタフェースが挙げられる。具体的には、図1のように2コマ目の認識の画像選択欄を横並びのスライド形式にし、横並びのスライドをループできるようにすることで、半側空間無視を有する人でも使えるようにしている。



図 1 コンテンツのユーザーインタフェース

1.4. 高次機能障害

高次機能障害は、いわゆる単症状としての失語・失行・失認のほかに行動障害や注意障害、遂行機能障害、社会的行動障害などの後遺症の総称である。^[1] 単症状とは、脳の限局した一部の破壊によって現れる症状である。^[2] 主な発症原因は、脳血管障害や頭部外傷などによる脳の器質的損傷であり、障害の現れ方は損傷部位や損傷の程度によって異なる。

また、高次機能障害の症例は以下の通りである。^[1]

症状	症状の例
記憶障害	物の置き場所を忘れる。 新しい出来事を覚えられない。 同じことを繰り返し質問する。
注意障害	ぼんやりしていて、ミスが多い。 二つのことを同時に行うと混乱する。 作業を長く続けられない。
遂行機能障害	自分で計画を立てて物事を実行することが出来ない。 人に指示してもらわないと何もできない。 約束の時間に間に合わない。
社会的行動障害	興奮する、暴力を振るう。 重い道理にならないと、大声を出す。 自己中心的になる。

1.5. 認知行動療法

認知行動療法（CBT : Cognitive and Behavioral Therapies）とは、認知に働きかけることで気持ちを楽にする精神療法・心理療法の一種である。^[3] ここで言う「認知」とは、出来事などに対する患者内の思考、態度、信念などを意味する。^[4] 認知行動療法の目的は、ストレスによる悲観的な考え方や、問題を解決できない心理的状态になるような考え方に対し、ストレスに対応することが出来る心理的状态を作ることである。^[5]

日本認知・行動療学会は、認知行動療法を次のように説明している。

- (1) 問題を具体的な行動（思考、情緒、運動すべてを含む精神活動）としてとらえ、どのような状況でそのような精神活動が生じるのかという行動分析をします。
- (2) 問題解決のための治療目標を具体的に明確にし、その変容を目指します。
- (3) 「行動」の変容のためには、どのような体験が必要であるかという考え方をします。
- (4) 観察可能あるいは定義可能なパラメータ（例えば、点数化した不安の程度、ある行動の頻度や持続時間、脈拍などの生理学的な測定）を用いて、試行する治療の効果を検証することができます。
- (5) 問題や疾患に応じた治療プログラムが多くつくられており、それらの多くで治療効果が実証されています。

（日本認知・行動療学会 2019/8）^[6]

上記の説明では、患者の抱える問題がどのような行動でどのように精神活動が発生するのかを分析し、各患者の行動を変化させるために患者に合わせた治療方法を適用する心理療法とされている。つまり、認知行動療法は、すべての患者に特定の治療方法を適用するのではなく、患者の精神活動や観測・測定されたパラメータを元に各患者に合わせた治療方法を適法する心理療法である。

この認知行動療法が、精神分析や森田療法といった他の心理療法と異なる点は、1 人の創始者がいるわけではなく、創始者が打ち立てた理論があるわけでもないことである。この「認知行動療法」という名前は、米国、英国、南アフリカなどの各地域で研究者が基礎的な心理学の原理や、研究成果や臨床実験から見出した理論に基づいて考案された心理療法を集約したものである。^[6]

認知行動療法の「行動療法」という名称は「行動を対象にした治療法」ではなく、この技法が行動理論（学習理論）を基盤としていることに由来する。^[6] 認知行動療法を構成する理論や原理には行動理論（学習理論）、認知理論、これらを発展させた理論が存在し、本研究では、認知理論を元とした認知療法や論理療法の考え方を参考に認知行動療法のアプリ開発を行っている。

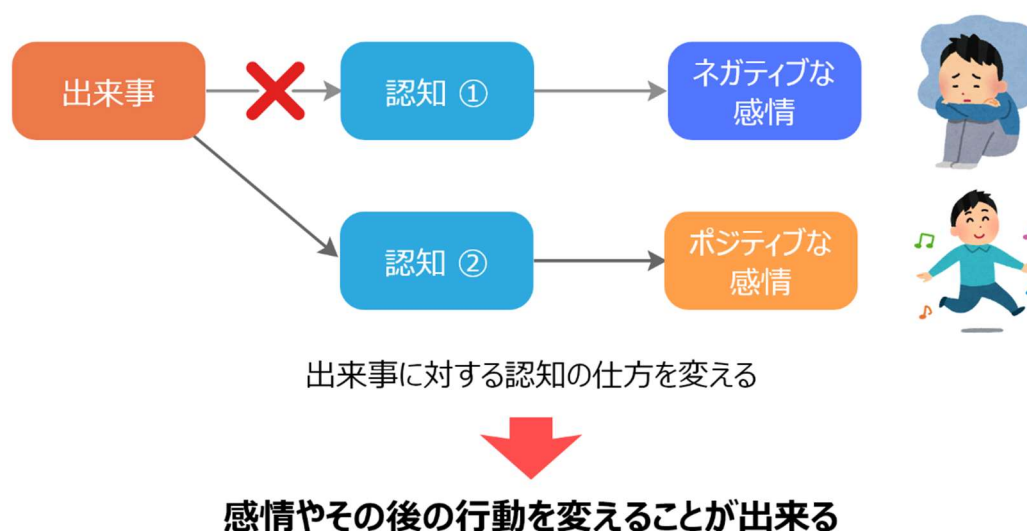


図 2 認知行動療法の図解

1.5.1. 認知療法

認知行動療法の一つとして認知療法がある。認知療法は、1963年にアーロン・ベック（Aaron T. Beck）によって提唱された心理療法であり、出来事に対する自動思考や認知の歪みに焦点を当て修正をすることで認知の歪みに起因する症状などを軽減することを目的とする短期精神療法の一つである。

主に、患者の偏った物事のとらえ方（認知）を修正させることで、より柔軟的で現実的な考え方や行動ができるように手助けをする療法である。

対象の症例として、うつ病が挙げられる。うつ病の患者は自身や周囲、将来に対して否定的・悲観的に考えてしまう心理の癖が心理的に悪影響を及ぼしてしまう。そのため、認知療法では、考え方を修正することで症状の改善を図る。^[7]

1.5.2. 論理療法

認知行動療法や認知療法の一つとして論理療法がある。論理療法は、1958年にアルバート・エリス（Albert Ellis）によって提唱され、その核となる理論が、ABC理論である。この理論では感情が発生するまでのプロセスを、「A（Activating event）：感情や行動を活性化させる刺激となるような出来事・状況」、「B（Beliefs）：出来事に対する評価・解釈・信念」、「C（emotional and behavioral Consequences）：最終的に出来事によってもたらされる感情・行動面の結果」として分けて考える。

このように3つの要素として分割したときに、A, B, Cの順で感情が発生していると考えられ、同じ出来事が発生しても認知（信念・解釈）の仕方によって感情が変化することを示している。^[8] すなわち、出来事が直接的に感情を引き起こすのではなく、認知の仕方を変えることによって感情や行動が変化するという理論である。そのため、論理療法はABC理論を元に、認知に働きかけることで感情や行動を変容させることを目的とした心理療法である。

2. 使用技術

アプリケーション開発に用いたプログラム言語やライブラリ、フレームワーク、その他使用技術は以下の通りである。

2.1. プログラム言語

主要開発言語として、JavaScript と Node.js を使用しており、API などのバックエンド言語として PHP を用いている。使用している言語の説明は以下の通りである。

2.1.1. JavaScript

JavaScript は、ウェブページにて動作する軽量なインタプリタ型のスクリプト言語である。同期的なコンテンツの書き換えやマルチメディアの管理、計算処理など多くのことを行うことが出来る。^[9] 本研究の主要開発言語の一つである。

2.1.2. Node.js

Node.js は、フロントエンドで動作する JavaScript をサーバーサイドや CLI（コマンドラインインタフェース）で動作させるようにした JavaScript の実行環境である。Node.js の最大のメリットは、フロントエンドとサーバーサイドのコードを同じ言語で記述することができ、サーバーサイドの学習が比較的容易になる点である。本研究の主要開発言語の一つであり、VueCLI と併用して HTML、CSS、JavaScript ファイルを生成している。

2.1.3. PHP

PHP は、広く使われているオープンソースのスクリプト言語である。主に Web 開発に用いられることが多く、HTML 形式の記述にバックエンドで動作するスクリプトを埋め込むことが出来る点が特徴である。HTML にスクリプトを埋め込む際には特別な開始および終了命令である「<?php」と「?>」で囲む必要がある。PHP の名前は、Hypertext Preprocessor を再帰的に並べたときの略称である。^[10] 本研究では、漫画の総数や 2 コマ目のページ数、タイトルなどの情報を取得する API として用いている。

2.2. ライブラリ・フレームワーク

Vue.js のエコシステムを用いて SPA（シングルページアプリケーション）のウェブアプリケーションを作成した。Vue.js のエコシステムを構築するには、VueCLI を用いている。

2.2.1. Vue.js

Vue.js または Vue は、Evan You が開発した、Web アプリケーションにおけるユーザーインタフェースを構築するための、オープンソースの JavaScript フレームワークである。Vue.js の最大の特徴は、控えめなリアクティブシステムという点である。^[11] 具体的にはデータの変更を検知し、そのデータの変更に対して即応的（リアクティブ）に表示を反映させることが出来る。特に、「データ変更に応じて自動的に表示が変わる」ことが重要であり、JavaScript 側から直接 DOM を操作するスクリプトを記述する必要がない点が jQuery などのライブラリとの違いである。

本研究では、データベースなどから取得したデータと表示を連動させる処理にリアクティブシステムを用いている。

2.2.2. VueCLI

VueCLI は、Vue.js でアプリケーションを開発するためのコマンドラインツールである。Node.js の環境で使うことが出来る、npm（Node Package Manager）コマンドを使用してパッケージをインストールする。「.vue」拡張子のコンポーネントファイルを webpack などのバンドラーでバンドルすることで HTML、CSS、JavaScript を出力する。VueCLI を用いる際のメリットとして、環境作成・設定が容易であるという点が挙げられる。Vue.js の環境構築に関する詳しい知識が無くても、コマンド 1 つと数個の設定項目を入力するだけで Vue.js の環境構築することが出来る。

VueCLI の「CLI」は、Command Line Interface の略で、コマンドライン上でシステムが動作することを意味する。

2.2.3. Firebase

Firebase は、mBaaS の一つであり、iOS や Android、ウェブなどのアプリケーションの開発に活用できるクラウドプラットフォームである。^[11] Firebase は、バックエンドで実行するユーザー認証やデータベースなどを提供しているため、開発者がそのような機能を用意する必要がなく、高機能なアプリケーションを開発することが出来る。Firebase の最大の特徴は、リアルタイム同期する認証やデータベースを使用できる点である。認証やデータベースのデータが変更されると、フロントエンド側のイベントが呼び出され、データの変更をアプリケーションの表示に反映させることが出来る。

mBaaS とは、Mobile Backend as a Service の略であり、スマートフォン向けのアプリケーションが必要とするサーバー側の様々な機能を一括して提供するクラウドサービスである。これにより、開発者はサーバーを設置・運営することなくアプリケーションを開発できる。^[13]

本研究では、以下の機能を用いてアプリケーションを構成している。

- Authentication

ユーザーの認証を行う機能である。認証方法は、E メールや電話番号の他に、Google や Facebook、Twitter のようなプロバイダとの認証連携がある。

本研究では、E メールと Google、匿名でアカウント認証を行っている。

- Realtime Database

データの保管と同期を行うことが出来る、NoSQL のクラウドデータベースである。データは JSON 形式で保存され、専用のルールを記述することで、データの読み込み・書き込みの権限を設定することが出来る。^[14]

本研究では、ユーザーやグループなどの情報やユーザーのログを保存するために用いている。

- Storage

写真や動画などのファイルを保管、提供することが出来るストレージサービスである。

^[15]

本研究では、ユーザーのプロフィール画像の保存先として利用している。

2.2.4. Flicking

Flicking は、HTML 内にある複数の横並びの要素をスワイプで選択できるようにする JavaScript ライブラリである。ユーザーインタフェースだけでなく、JavaScript 側からもスライド変更、スライドの変更情報の取得、各種設定の変更などを行うことが出来る。

本研究では、認知行動療法の認知に関する画像を選択するユーザーインタフェースを実装するために用いている。

2.2.5. Bootstrap Icons

Bootstrap Icons は 1500 以上のアイコンを備えたオープンソースのライブラリである。^[16]

本研究では、メニューの欄にアイコンを用いてそのリンク先の内容について視覚的に表現するために用いている。

2.3. 使用技術

Vue.js は、デフォルトで SPA 機能を備えている。また、オプションとして PWA 機能を組み込むことが出来る。

2.3.1. SPA

SPA とは、Single Page Application (シングルページアプリケーション) の略で、単一のウェブ文書を読み込み、別のコンテンツを表示するときに XMLHttpRequest や Fetch などの JavaScript API を用いて単一文書の内容を更新するアプリケーションである。

これにより、ユーザーはサーバーから新しいページ全体を読み込まずにウェブサイトを使用することが出来るため、シームレスなウェブアプリケーションを作成することが出来る。しかし、

SEO で不利になる，アプリケーションのダウンロードに時間がかかるなどのデメリットもある．
主要な SPA フレームワークとして，React, Angular, Vue.js がある．^[17]

2.3.2. PWA

PWA とは，Progressive Web Apps（プログレッシブウェブアプリ）の略で，iOS / Android などのネイティブアプリケーションの機能を備えた Web アプリケーションのことを指す．具体的には，アプリのインストール，オフラインでの動作，ホーム画面などのアイコンからアクセス可能にするなどの利点が挙げられる．

補足として，PWA という用語は，ウェブ技術のみで柔軟かつ適応性のあるアプリケーションを作成するというコンセプトのもと Google が当初使用していた略語であり，公式または正式な名称ではない．^[18]

2.4. 使用ツール・サービス

開発プラットフォームとして Git と GitHub を用いている．それぞれ機能を以下に示す．

2.4.1. Git・GitHub

Git は，小規模から非常に大規模なプロジェクトまで，効率的に処理するように設計された，オープンソース分散管理バージョン管理システムである．^[19]

GitHub は，ソースコードをホスティングすることで，他の開発者とのコードレビューや，プロジェクトの管理をしつつソフトウェアの開発を行うことが出来る開発プラットフォームである．^[20]

本研究では，ソースコードの開発・管理や，GitHub Actions と GitHub Pages を用いた自動デプロイに用いている．

3. 研究内容

3.1. アプリケーションの概要

3.1.1. アプリのフローチャート

ユーザーがサイトにアクセスする際に、ユーザーは **Firebase** の認証機能を通してユーザー認証する。ユーザー認証後は認証時の情報を元に、ユーザー情報を **Realtime Database** に保存する。

ユーザーが漫画を閲覧する場合は、漫画のタイトルや漫画数などの情報をヘテムルサーバーにある **PHP** プログラムから取得する。その情報を元にアプリケーション側が **URL** を自動で生成し、漫画の画像を表示する。閲覧を開始した時と終了した時にログをデータベースに送信する。

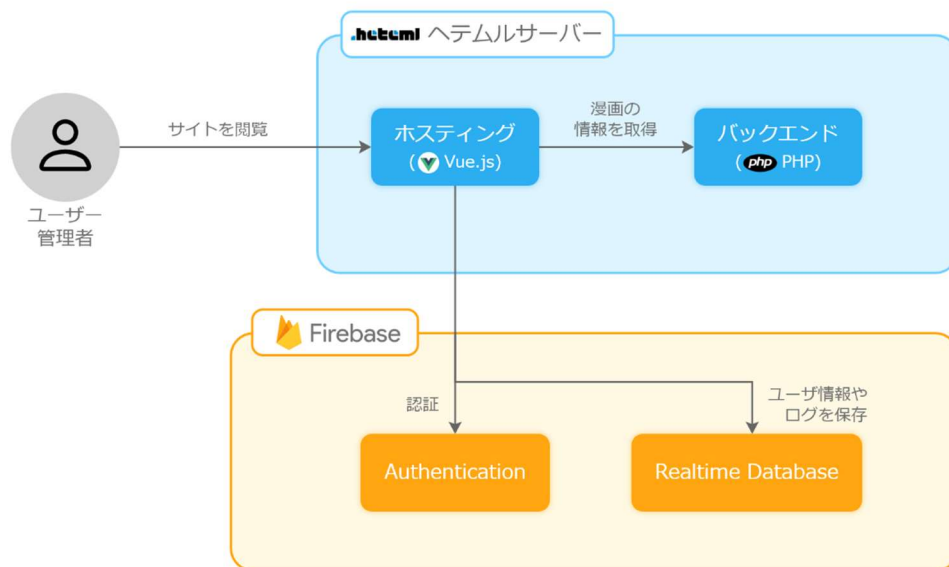


図 3 システム構成フローチャート

3.1.2. ユーザーと管理者

本システムでは、認証する利用者をユーザーと管理者に分けている。本システム上で行う通常の認証方法では、ユーザーのみ登録可能である。管理者の登録方法は、**Firebase Console** から行う。**Realtime Database** の **admin-users** 内に **true** の真偽値を持つ認証されているユーザーID を追加することでユーザーを管理者として登録することが出来る。

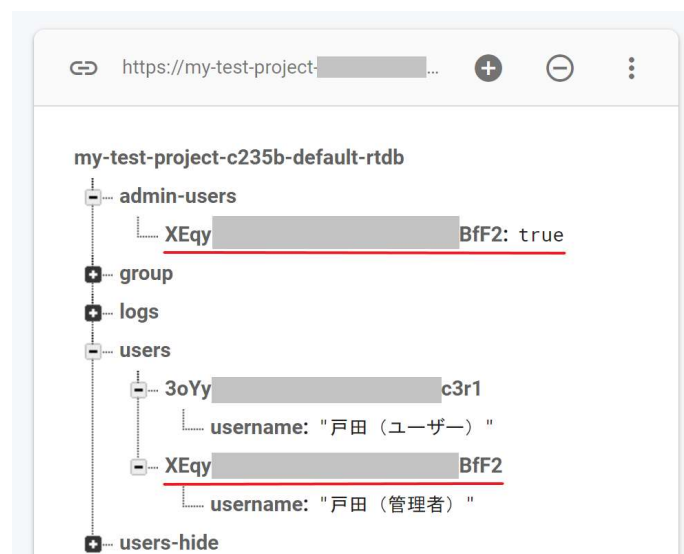


図 4 Realtime Database の管理者指定方法

3.1.3. 漫画コンテンツ

本研究で使用する漫画コンテンツについて、当研究室の野田氏と協力し、野田氏に作成していただいたストーリー、画像を使用している。

漫画コンテンツは 3 つのセクションに分かれており、それぞれ、出来事、認知、感情を表現している。また、認知・感情のセクションでは、認知のパターンの数だけ画像を用意している。

3.2. 開発環境の構成

本システムのファイル構成や使用しているオプションについて説明する。

3.2.1. ディレクトリ構成

本システムの開発環境において、VueCLI で生成されたディレクトリ構成の説明は次の通りである。

- **node_modules**

Node.js でインストールしたモジュールが保存されるフォルダである。npm install コマンドなどを用いて、モジュールをインストールできる。

- **dist**

dist は、分配や配布などの意味である distribution の略で、Web アプリケーションのとして配布するための出力コードを意味する。npm run build コマンドによってビルドされた HTML/CSS / JavaScript などの出力ファイルを保存するためのフォルダである。

- **public**

public は、Web アプリケーションとして公開する際のディレクトリ構成の元となるフォルダである。画像や Web サイト用の設定ファイル、別途に用意した HTML ファイルなどがある場合は、このフォルダに格納することで出力先の dist フォルダに反映される。

- **src**

src フォルダは、サイトを構成するファイルを格納している。src フォルダ内にある App.vue ファイルをベースとして、各コンポーネントファイルを読み込ませてアプリケーションを構成している。

- **tests**

テストコードを格納するフォルダである。本システムでは、汎用関数などの JavaScript ファイルのテストコードが格納されている。

本システムのテスト環境は Jest であり、このフォルダに格納されたテストファイルは npm run test コマンドで実行する。

3.2.2. ルーティング

本システムのルーティング（ページ遷移）には、Vue Router を用いている。Vue Router は Vue.js 公式のルータであり、Vue エコシステムを用いたシングルページアプリケーションの構成でルート機能を実装するために用いる。

Vue Router のルート・マッピングの設定は、src / router フォルダ内の index.js ファイルで指定する。本システムの設定ではサイトの安定性を考え、サイトの URL に#（ハッシュ）を挿入することでルーティングを制御する hash モードを使用している。

3.2.3. 状態管理

本システムの状態管理パターン・ライブラリとして Vuex を用いている。これは、ゲッターのような予測可能な方法によってのみ状態の変異を行うというルールを保証し、アプリケーション内のすべてのコンポーネントのための集中型のストアとして機能する。^[21]

Vue.js では、各コンポーネントが変数や関数を有しているが、コンポーネント間で変数や関数を受け渡しすることは難しい。そのため、Vuex を用いて共通の変数を使うことでデータの受け渡しを可能にしている。

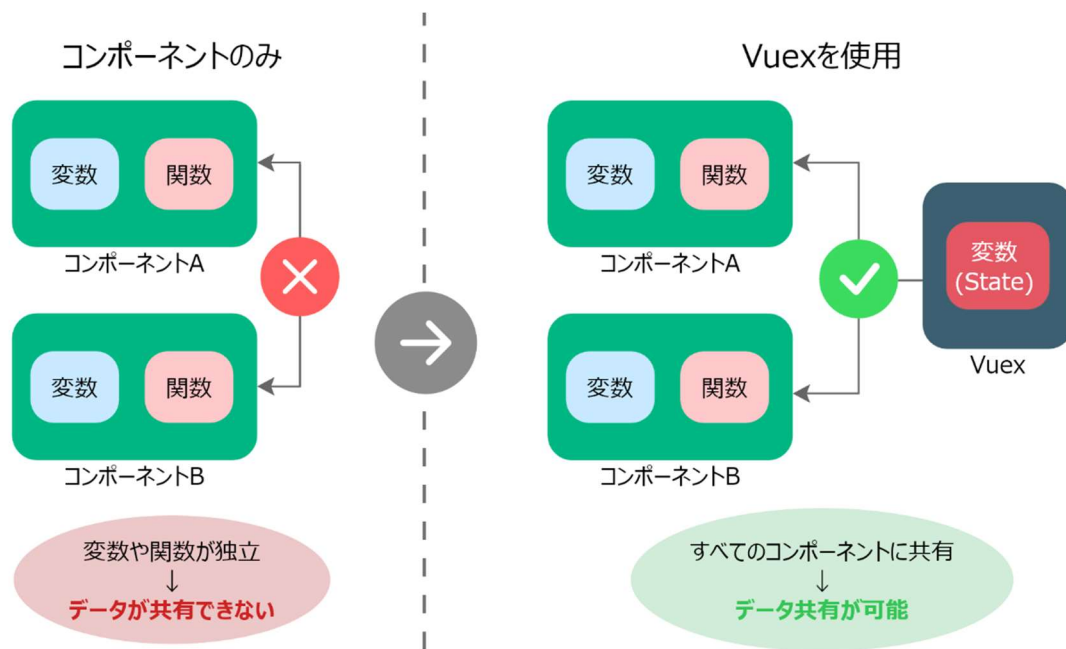


図 5 Vuex の導入のメリット

3.3. コンポーネント

本システムで作成した各コンポーネントについて説明する。

3.3.1. ヘッダー

ヘッダーには、ホーム画面にリンクされたタイトルと、ログインボタン、ユーザーアイコンを配置している。

ログインボタンとユーザーアイコンはログイン状態によって表示を変えており、ログインしていない状態の際はログインボタンを表示し、ログインしている状態の際はユーザーアイコンを表示する。ログイン状態の判別は、Vue Router で管理されているログイン状態を参照している。

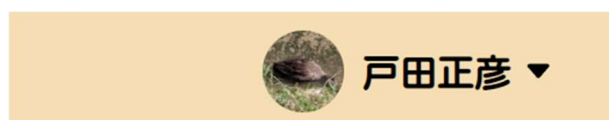


図 6 ログイン済み状態のヘッダー



図 7 ログインしていない状態のヘッダー

ログインしている状態でユーザーアイコンおよびユーザー名をクリックすることで、ユーザーメニューを表示できる。ユーザーメニューには、ユーザー設定やホームなどのリンクや、ユーザーメニューはユーザーメニュー外をクリックすることで非表示にすることが出来る。

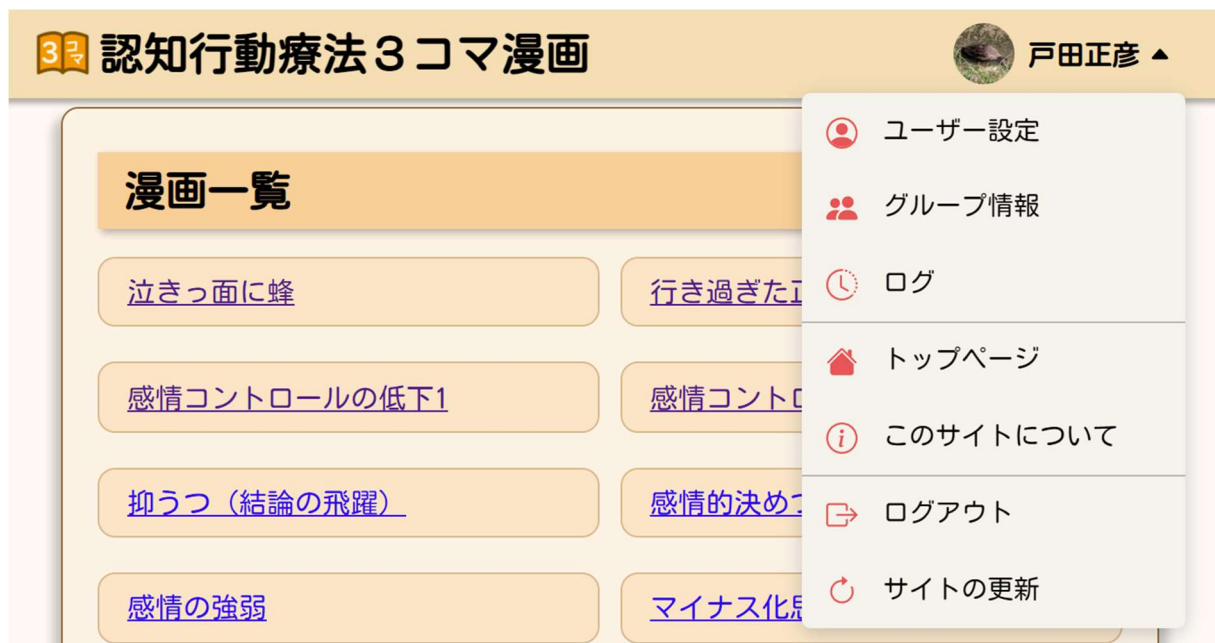


図 8 ユーザーメニュー

3.3.2. フッター

フッターには、関連した各種リンクを配置している。現在は、開発におけるリンクを配置しているが、認知行動療法に関連したリンクを配置する予定である。

3.4. サイト構成

3.4.1. ログイン画面

本システムのログイン画面は以下の図の通りである。



図 9 ログイン画面

ログイン画面について、Firebase と FirebaseUI を用いて実装している。FirebaseUI は Firebase Authentication のユーザーインタフェースやログイン機能を備えたライブラリである。

- HTML

FirebaseUI を表示させるための HTML は下記の通りである。表示する DOM には JavaScript と共有の id を指定する。

```
<div id="firebaseui-auth-container"></div>
```

図 10 FirebaseUI - HTML

- JavaScript

FirebaseUI の設定は下記の通りである。

FirebaseUI のインスタンスを定義した上で、そのインスタンスの start メソッドを用いることで FirebaseUI の DOM を展開できる。引数には、表示する DOM の id を指定する。

signInSuccessUrl は、サインインが成功した際にリダイレクトする URL であり、本システムではホーム画面をリダイレクトする URL として指定している。

signInOptions は、ログインプロバイダの表示などの設定を指定する。本システムでは、メールと Google アカウント、ゲスト認証を許可しており、メール認証では確認メールを送信することでメールの無断使用を防いでいる。

callbacks には、UI 表示時に英語を日本語に変換する処理や、ログイン成功時に email と名前をデータベースに保存する処理のようなイベント処理を記述している。

```
ui.start("#firebaseui-auth-container", {
  signInSuccessUrl: "./",
  signInOptions: [
    {
      provider: firebase.auth.EmailAuthProvider.PROVIDER_ID,
      // ログイン時に確認メールを送信する
      signInMethod:
        firebase.auth.EmailAuthProvider.EMAIL_LINK_SIGN_IN_METHOD,
      forceSameDevice: false,
    },
    firebase.auth.GoogleAuthProvider.PROVIDER_ID,
    firebaseui.auth.AnonymousAuthProvider.PROVIDER_ID,
  ],
  callbacks: {
    /* 省略 */
  }
});
```

図 11 FirebaseUI - JavaScript

3.4.2. ホーム

ホーム画面には、ログイン後にコンテンツを閲覧することが出来るように、コンテンツ一覧を配置している。

3コマ

認知行動療法3コマ漫画

 戸田正彦 ▼

漫画一覧

[泣きっ面に蜂](#)

[行き過ぎた正義感](#)

[感情コントロールの低下1](#)

[感情コントロールの低下2](#)

[抑うつ（結論の飛躍）](#)

[感情的決めつけ](#)

[感情の強弱](#)

[マイナス化思考](#)

使用ライブラリ

[Flicking](#)
[Flicking Plugins](#)

GitHub

[three-panel-comic-demo](#)

研究室サイトリンク

[2021研究室](#)
[戸田の卒研サイト](#)

2021年 卒業研究

図 12 ホーム画面

コンテンツ一覧は、研究室で契約しているレンタルサーバーにある PHP プログラムから情報を取得して表示している。コンテンツの総数は `countComic.php` から取得し、各コンテンツのタイトルは、`getData.php` から取得している。

PHP の自作 API からコンテンツ情報を取得する方法として、`fetch` 関数を用いる。`fetch` 関数は、Web 上にあるデータファイルや API からデータを取得する `Promise` を戻り値とする非同期関数である。本システムでは、コンテンツ総数を取得した後にすべてのコンテンツのタイトルを取得するため、非同期処理である `fetch` 関数を、`async / await` を用いて同期的に処理を実行している。


```

async mounted() {
  await fetch(`${this.originalPath}countComic.php`)
    .then((res) => res.json())
    .then((data) => {
      this.comicLength = data.count;
    })
    .catch((e) => console.error(e));

  let linkArray = [];
  let titleArray = [];

  for (let i = 1; i <= this.comicLength; i++) {
    linkArray.push(`comics/${i}`);


    await fetch(`${this.originalPath}getData.php?comic=${i}`)
      .then((res) => res.json())
      .then((data) => {
        if (data.status === false) {
          console.error("fetch error: getData file throw error");
        } else {
          titleArray.push(data.title);
        }
      })
      .catch((e) => console.error(e));
  }
  this.links = linkArray;
  this.titles = titleArray;
}


```

図 13 コンテンツデータ取得コード

3.4.3. サイト情報

サイト情報には、本システムの目的や、認知行動療法についての解説を載せている。認知行動療法に対する理解や、本システムの目的を知ってもらうことを目的としている。

 **認知行動療法 3 コマ漫画**

 **戸田正彦** ▼

このサイトについて

認知行動療法とは

認知行動療法は、ものの考え方や受け取り方（認知）に対して働きかけることで、気持ちを楽しんだり、ストレスを軽減する治療方法です。

このサイトの目的

私たちの感情は、ものの考え方や受け取り方（認知）によって左右されます。

認知が変われば感情が変わり、いずれ行動の変化につながります。

このサイトはそのような認知と感情の結びつきについて学んでもらい、自身の認知について気づいてもらうことが目的です。

図 14 サイト情報

3.4.4. 漫画コンテンツ

漫画コンテンツは、左側に簡易な説明文章、右側にコンテンツ画像を表示している。コンテンツは、認知行動療法の出来事・認知・感情の3つのセクションに分けている。

1 コマ目は、起こった出来事のスライドを表示している。

2 コマ目は、出来事に対する認識のスライドを表示しており、複数の認識のパターンを選択する方法をとっている。認識を選択する方法は、Flicking ライブラリを用いており、スライドをボタンもしくはスワイプすることで別の選択肢のスライドを選択できる。

3 コマ目は、出来事に対する認識によって最終的にもたらされる感情のスライドを表示しており、認識の選択肢に応じた感情が表示されるようにしている。感情のスライドを動的に変更する方法は、Vue.js ライブラリのデータバインディング機能を用いる。データバインディング機能とは、データの変更が表示 (View) に自動的に反映させる機能である。上記の機能を用いて、2 コマ目の選択肢が変更された際に、3 コマ目の画像の URL を生成し、そのデータを表示側の画像の src 属性に反映させている。

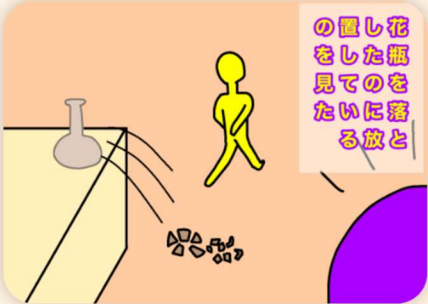
3冊 認知行動療法 3 コマ漫画

戸田正彦 ▼

マイナス化思考


出来事


起こった出来事のスライドです。




認識

右のスライドを動かして選んでください。







1/3

感情

認識に対応したスライドが表示されます。

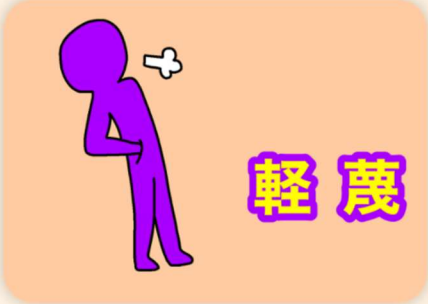


図 15 漫画コンテンツの画面

3.4.4.1. Flicking

本システムの Flicking ライブラリのオプションは以下の通りである。

プロパティ	値	説明
autoInit	false	Flicking が読み込まれたときに自動で init メソッドを実行しないようにする。 init メソッド後に画像を読み込むと想定しない配置になるため、手動で init メソッドを実行するために autoInit を無効化する。
align	true	選択されたスライドを中心に配置する。
circular	true	横並びのスライドをループさせる。 右か左の片側のどちらかが見えなくなる半側空間無視を有する人を考慮して有効にする。
deceleration	0.5	スライドをスワイプで移動させた場合のスライドの減速値。 デフォルトより大きくすることで、スライドを選択しやすいようにする。
duration	350	スライドをスワイプで移動させた場合にスライドが停止するまでの時間（ミリ秒）。 通常の 500 ミリ秒よりも短くすることで、スライドを選択しやすいようにする。
renderOnlyVisible	true	表示幅から外れたスライドをレンダリングせず、表示しない設定。 アプリの快適度を高めるために有効化する。

```
<Flicking
  class="flex-right-item"
  ref="flicking"
  :options="{
    autoInit: false,
    align: 'center',
    circular: true,
    deceleration: 0.05,
    duration: 350,
    renderOnlyVisible: true,
  }"
  :plugins="plugins"
  @changed="changePanel"
>
  <div
    v-for="(url, index) in imageUrl.second"
    :key="index"
    :id="'page-{{index}}'"
    class="panel comic-page"
  >
    
  </div>
  <span slot="viewport" class="flicking-arrow-prev is-circle"></span>
  <span slot="viewport" class="flicking-arrow-next is-circle"></span>
  <div slot="viewport" class="flicking-pagination"></div>
</Flicking>
```

図 16 Flicking ライブラリの HTML

Vue.js での Flicking ライブラリの実装は以下のように、Flicking ライブラリをインポートした上で components プロパティに Flicking を記述する。Flicking ライブラリのプラグインを使用する場合は、必要なプラグインをインポートした上で、data 内の plugins プロパティ内に各要素がプラグインのインスタンスとなる配列を記述する。

```
import { Flicking } from "@egjs/vue-flicking";
import { Arrow, Pagination } from "@egjs/flicking-plugins";

export default {
  data() {
    return {
      /* --- 省略 --- */
      plugins: [new Arrow(), new Pagination({ type: "fraction" })],
    };
  },
  components: {
    Flicking: Flicking,
  },
};
```

図 17 Flicking ライブラリの JavaScript

3.4.5. ユーザー設定

ユーザー設定ページでは、ユーザー情報の編集やログアウトをすることが出来る。編集可能な情報は、ユーザー名とアイコン画像であり、メールアドレスは変更できない仕様になっている。ユーザー名の入力については、一部の NG ワードを入力できないようにしている。アイコン画像については、画像のデータを **Firebase Storage** に保存し、その画像のアクセス URL を **img** タグの **src** 属性とすることで表示している。ログアウトについては、**Firebase Authentication** の機能を用いてログアウトしている。

ユーザー情報は、**Vuex** ストアから取得している。**App.vue** が読み込まれた際に **Firebase** からユーザー情報を読み込み、**Vuex** ストアに格納している。

図 18 ユーザー設定の画面

NG ワードを検知するコードの説明は以下の通りである。

util フォルダ内に **checkNgWords.js** として格納しており、関数のモジュールとして機能する。この関数は、同フォルダ内にある **ngWords.json** から NG ワードデータを取得し、引数の文字列が NG ワードを含んでいるかを判別する。

文字列の判別方法には **match** メソッドを用いており、関数の戻り値として真偽値を返す。

```
import ngWords from "./ngWords.json";

/**
 * 文字列内にNGワードが含まれているかチェックする
 * @param {String} str チェックする文字列
 * @returns {Boolean} true または false
 */
function checkNgWords(str = "") {
  let flag = false;

  ngWords.data.forEach((ngWord) => {
    if (str.match(ngWord)) {
      flag = true;
    }
  });

  return flag ? true : false;
}

export default checkNgWords;
```

図 19 NG ワードを検知するコード

3.4.6. グループ情報

グループ情報ページでは、ユーザーと管理者によって表示が異なる。ユーザーの場合は、「グループ情報」のみ表示される。一方で管理者の場合は「グループ情報」に加え、「メンバーリスト」と「グループの編集」も表示される。

ユーザーの場合、グループ情報画面は以下の通りになる。グループの情報のみしか表示されない。

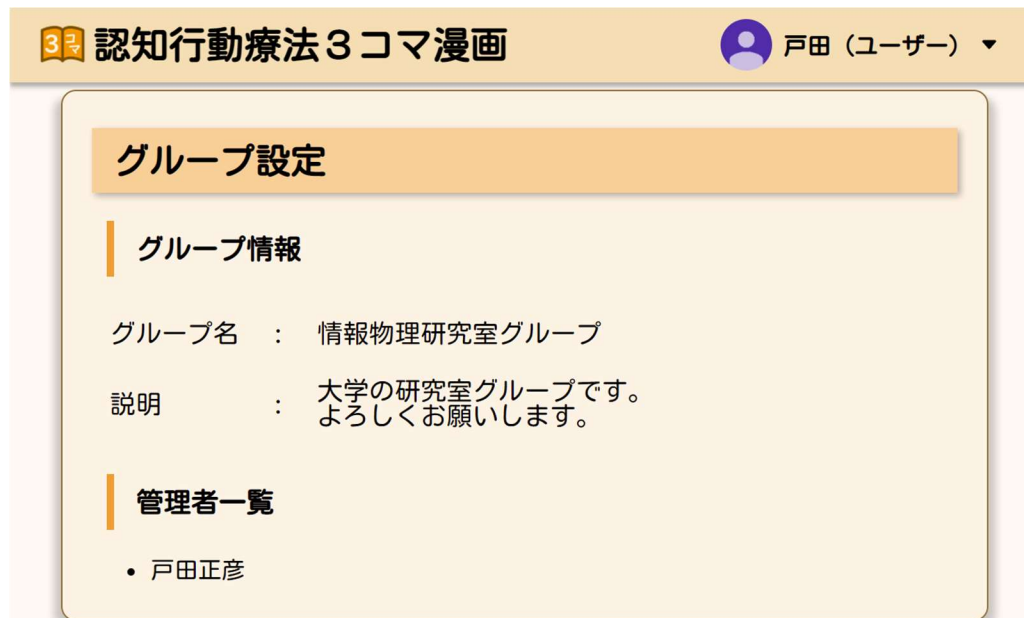


図 20 ユーザーでのグループ情報画面

管理者の場合、グループ情報画面は以下の通りになる。グループの情報の他に、メンバーリストとグループの編集へのナビゲーションリンクがある。

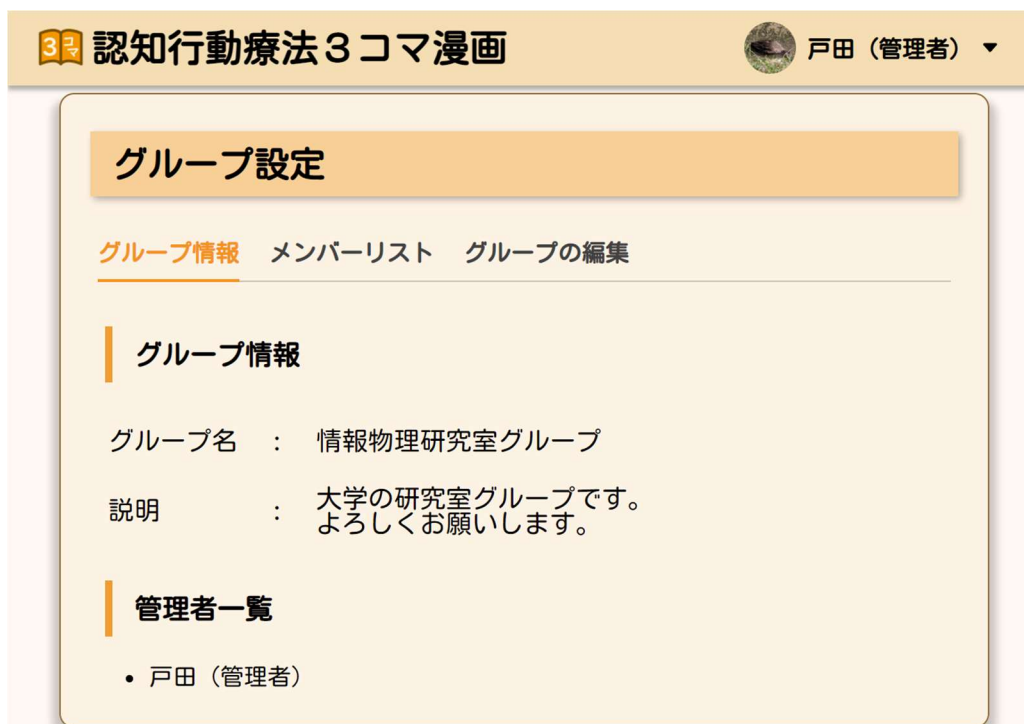


図 21 管理者でのグループ情報画面

3.4.6.1. グループ情報

グループ情報画面では、グループ情報と管理者一覧が表示される。この情報は Firebase の Realtime Database に保存されているデータから読み込んでいる。

認知行動療法 3 コマ漫画 戸田 (管理者) ▼

グループ設定

グループ情報 メンバーリスト グループの編集

グループ情報

グループ名 : 情報物理研究室グループ

説明 : 大学の研究室グループです。
よろしくお願いします。

管理者一覧

- 戸田 (管理者)

図 22 グループ情報の画面

3.4.6.2. メンバーリスト

メンバーリスト画面では、管理者を含めたすべてのユーザー情報の一覧が表示されている。このリストに表示されるユーザー情報は、ユーザー名、メールアドレスである。

また、リストの右端にある「ログを確認する」ボタンを押すことによって、そのユーザーのログを表示することが出来る。ログを確認できるようにした理由は、ユーザーのアプリケーションの実行状況を管理者が確認することによって、より確実に本システムを使ってもらうためである。

リスト上部にある入力欄は、ユーザーを検索するためのものであり、入力した文字列と部分一致したユーザー名を持つユーザーのみを表示させることが出来る。

認知行動療法 3 コマ漫画 戸田 (管理者) ▼

グループ設定

グループ情報 **メンバーリスト** グループの編集

メンバーリスト

以下の入力欄でユーザーを検索できます。

ユーザー名	メール	ログ
戸田 (ユーザー)	18*****@hiro.kindai.ac.jp	ログを確認する
戸田 (管理者)	ms*****an@gmail.com	ログを確認する

図 23 メンバーリストの画面

3.4.6.3. グループの編集

グループの編集画面では、ユーザーに表示するグループ情報を編集することが出来る。編集できる情報は、グループ名とグループ説明である。

3 認知行動療法3コマ漫画

戸田（管理者） ▼

グループ設定

グループ情報 メンバーリスト グループの編集

グループの編集

グループの情報を編集できます。

グループ名

情報物理研究室グループ

グループ説明

大学の研究室グループです。
よろしくお願いします。

変更

図 24 グループの編集の画面

3.4.7. ログ

ログの画面では、ユーザーが閲覧した漫画コンテンツの履歴や、コンテンツ全体を通した進行度表を表示している。

進行度表と履歴に分けてそれぞれ説明する。

ログ

戸田（管理者）さんのログ

進行度表

タイトル	進捗	閲覧数
泣きっ面に蜂	終了済み	19
行き過ぎた正義感	終了済み	12
感情コントロールの低下1	終了済み	3
感情コントロールの低下2	未完了	0
抑うつ（結論の飛躍）	終了済み	1
感情的決めつけ	終了済み	1
感情の強弱	終了済み	1
マイナス化思考	終了済み	1

ソートする項目：

日時

並び順： ☐ 昇順 ☒ 降順

漫画ID: 1
リンク: [泣きっ面に蜂](#)
実行状況: 終了済み
開始日時: 2022年 2月1日 21:20 48秒
終了日時: 2022年 2月1日 21:20 51秒

漫画ID: 8
リンク: [マイナス化思考](#)
実行状況: 未完了
開始日時: 2022年 1月31日 15:52 36秒

図 25 ログの画面

3.4.7.1. 進行度表

進行度表では、各漫画コンテンツに対するユーザーの進捗状況や今までの閲覧回数を表示している。この表は、Firebase の Realtime Database から取得したログの一覧から、閲覧回数を filter 関数で実行済みのログのみカウントしている。カウントした閲覧回数を、進捗状況に反映させている。

タイトル	進捗	閲覧数
泣きっ面に蜂	終了済み	19
行き過ぎた正義感	終了済み	12
感情コントロールの低下1	終了済み	3
感情コントロールの低下2	未完了	0
抑うつ（結論の飛躍）	終了済み	1
感情的決めつけ	終了済み	1
感情の強弱	終了済み	1
マイナス化思考	終了済み	1

図 26 ログの進行度表

以下のコードは実行済みのログを数える JavaScript コードである。

漫画の ID を引数とし、すべてのログに対して実行済み且つ漫画 ID と一致するログの配列を作成している。戻り値として、上記の方法で得た配列の要素数を返している。

```
countComicViews(comicId = "") {  
  const filteredArray = this.logs.filter((item) => {  
    return item.done === true && item.comicId === comicId;  
  });  
  
  return filteredArray.length;  
},
```

図 27 指定の漫画の閲覧数を取得するコード

3.4.7.2. 履歴

履歴では、ユーザーのログ情報をリストで表示している。表示している情報は以下の通りである。

- 漫画 ID
- リンク
- 実行状況
- 開始日時
- 終了日時

実行状況は、ユーザーが漫画コンテンツのすべての選択肢を閲覧した場合に終了済みになり、それ以外の場合は、未完了となる。

上部にある入力項目は、履歴のソートの項目や表示順を変更するためのものである。

ソートする項目：

日時

並び順： ☐ 昇順 ☒ 降順

漫画ID: 1
リンク: [泣きっ面に蜂](#)
実行状況: 終了済み
開始日時: 2022年 2月1日 21:20 48秒
終了日時: 2022年 2月1日 21:20 51秒

漫画ID: 8
リンク: [マイナス化思考](#)
実行状況: 未完了
開始日時: 2022年 1月31日 15:52 36秒

漫画ID: 8
リンク: [マイナス化思考](#)
実行状況: 未完了
開始日時: 2022年 1月31日 15:33 0秒

漫画ID: 7
リンク: [感情の強弱](#)
実行状況: 終了済み
開始日時: 2022年 1月31日 15:31 43秒
終了日時: 2022年 1月31日 15:31 50秒

漫画ID: 5

図 28 ログの履歴

4. まとめ

本研究では、認知行動療法 3 コマ漫画アプリケーションの開発を目的として開発に取り組み、Firebase や Vue.js などの技術を取り入れたアプリ開発をした。結果としてユーザーテストを行うことはできなかったが、PWA 対応やログの確認機能など、多くの機能を備えたアプリケーションにすることが出来たと考える。また、漫画コンテンツの部分では、スライドに関するライブラリを導入することで、一般の人だけでなく、高次機能障害などの障害を有している人にも対応したユーザーインターフェースを備えることが出来た。

4.1. 改善点

本システムの改善点は以下の通りである。

- ユーザーの実証実験を行っていない点。
開発期間が長くなってしまい、ユーザーテストを行うことが出来なかった。実際に使ったあとのフィードバックから改善点を見つけるべきだと考える。
- Function テストコードは作成したが、ユニットテストコードを作成していない点。
各コンポーネントにおける表示や API の呼び出しなどのテストコードを作成することが出来なかった。開発期間に余裕を持たせる必要があった。

5. 謝辞

本論文の作成にあたり、指導教員の徐丙鉄教授から、多大なご指導を賜りました。ここに感謝の意を表します。

また、心理学ミーティングに参加していただいた皆様には本システムの助言をして下さり、アプリ開発に協力いただいたことに改めて感謝致します。

6. 参考文献

- (1) 高次脳機能障害を理解する | 国立障害者リハビリテーションセンター (2022/1/17)
http://www.rehab.go.jp/brain_fukyu/rikai/
- (2) 巣症状とは - コトバンク (2022/1/31)
<https://kotobank.jp/word/%E5%B7%A3%E7%97%87%E7%8A%B6-1355438>
- (3) 橋本 優花里, 澤田 梢, 鈴木 伸一 (2006) 高次脳機能障害における認知行動療法の適用について p24~25 (2022/1/19)
https://fukuyama-u.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=5611&item_no=1&page_id=31&block_id=65
- (4) 認知行動療法とは | 認知行動療法センターのご案内 | 認知行動療法センター (CBT センター) (2022/1/23) <https://www.ncnp.go.jp/cbt/guidance/about>
- (5) 認知行動療法 | 脳と心の科学について学ぼう (2022/1/19)
<https://www.fun.ac.jp/~hanada/kokoronokagaku/concept5.html>
- (6) 日本認知・行動療法学会 (2019/8) 認知行動療法事典 p2 (2022/1/24)
- (7) 認知療法 | e-ヘルスネット (厚生労働省) (2022/2/6)
<https://www.e-healthnet.mhlw.go.jp/information/dictionary/heart/yk-045.html>
- (8) 重松潤, 尾形明子, 伊藤義徳 (2018) 認知行動療法と「腑に落ちる理解」 広島大学心理学研究 p.136 (2021/1/19) <https://core.ac.uk/download/pdf/197311627.pdf>
- (9) JavaScript とは - ウェブ開発を学ぶ | MDN (2022/1/20)
https://developer.mozilla.org/ja/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- (10) PHP: PHP とはなんでしょう? - Manual (2022/1/20)
<https://www.php.net/manual/ja/intro-what-is.php>
- (11) リアクティブの探求 — Vue.js (2022/1/25) <https://jp.vuejs.org/v2/guide/reactivity.html>
- (12) Firebase とは? Google 社の強みを活かしたメリットや... | Udemy メディア (2022/1/26)
<https://udemy.benesse.co.jp/development/system/what-is-firebase.html>
- (13) BaaS (mBaaS) とは - IT 用語辞典 e-Words (2022/1/26) <https://e-words.jp/w/BaaS.html>
- (14) Firebase Realtime Database | Firebase Documentation (2022/1/26)
<https://firebase.google.com/docs/database>
- (15) Cloud Storage for Firebase | Firebase Documentation (2021/1/26)
<https://firebase.google.com/docs/storage>
- (16) Bootstrap Icons · Official open source SVG icon library for Bootstrap (2022/1/26)
<https://icons.getbootstrap.com/>
- (17) SPA (単一ページアプリケーション) - MDN Web Docs 用語集: ウェブ関連用語の定義 | MDN (2022/1/26) <https://developer.mozilla.org/ja/docs/Glossary/SPA>
- (18) プログレッシブウェブアプリの紹介 - プログレッシブウェブアプリ (PWA) | MDN (2022/1/26) https://developer.mozilla.org/ja/docs/Web/Progressive_web_apps/Introduction
- (19) Git (2022/1/26) <https://git-scm.com/>
- (20) GitHub | GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside millions of other developers. (2022/1/26) <https://github.co.jp/>
- (21) Vuex とは何か? | Vuex (2022/1/30) <https://vuex.vuejs.org/ja/>