**令和7年度**

**情報処理科　卒業制作**

**SpotMusic班**

・**目次**

1.プロジェクト概要 3

1.1[チームメンバー 3](#_heading=h.1fob9te)

[1.2 スケジュール 3](#_heading=h.3znysh7)

[1.3 環境 3](#_heading=h.2et92p0)

[・開発環境 3](#_heading=h.tyjcwt)

[・動作環境 3](#_heading=h.3dy6vkm)

2.[アプリケーション名 4](#_heading=h.4d34og8)

[2.2　アプリケーションコンセプト 4](#_heading=h.2s8eyo1)

[2.3　機能一覧 4](#_heading=h.17dp8vu)

[・ログインしていない状態でできること 4](#_heading=h.3rdcrjn)

[2.4　機能説明 4](#_heading=h.26in1rg)

[― ログイン画面 5](#_heading=h.lnxbz9)

[― 新規登録画面](#_heading=h.1ksv4uv)　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　 7

[― メイン画面](#_heading=h.44sinio) 8

[― プレイリスト操作](#_heading=h.2jxsxqh) [10](#_heading=h.2jxsxqh)

[― 曲の検索](#_heading=h.z337ya) 　　　 11

[― フォロー操作](#_heading=h.z337ya)　 　　 12

[― 再生プレイヤー 1](#_heading=h.z337ya)2

[―](#_heading=h.3j2qqm3) フレンドリスト画面　　 14

[―](#_heading=h.3j2qqm3) ユーザーリスト画面 　　　　 15

[―](#_heading=h.3j2qqm3) ブロックリスト画面 　　 15

[―](#_heading=h.3j2qqm3) チャット画面 　　 　 16

[―](#_heading=h.3j2qqm3) プレイリスト画面（コメント） 　　17

[―](#_heading=h.3j2qqm3) モバイルデバイス [1](#_heading=h.4i7ojhp)8

[―](#_heading=h.3j2qqm3) [参考サイト 21](#_heading=h.2xcytpi)

[―](#_heading=h.3j2qqm3) [参考文献 21](#_heading=h.1ci93xb)

[3.](#_heading=h.3whwml4) [起こった不具合 2](#_heading=h.2bn6wsx)2

[・良かった点](#_heading=h.qsh70q) 22

[・苦労した点](#_heading=h.1pxezwc) 22

[・改善点](#_heading=h.49x2ik5) 22

・個人コメント 23

1. **プロジェクト概要**
   1. チームメンバー

― 小池嶺馬

― THANH CHAU NGUYEN

― 齊野航平

— 佐藤択弥

― 田久保雅也

― 張宸嘉

― 横本晃大

* 1. スケジュール

期間：2024年11月初旬~2025年2月28日

・2024年11月初旬～

外部設計、データベース定義

・2024年12月中旬～

内部設計、デザイン、実装

* 1. 環境

　・開発環境

― OS：Windows11

― 言語：Java 11, HTML, CSS, JavaScript

― データベース：MySQL 8.0.35

― サーバー：Tomcat9, AWS EC2 Amazon Linux2 t2.micro

　・動作環境

― ブラウザ：Google Chrome

― 携帯端末からの利用は、一部機能のみ想定している

― Web ブラウザのバージョンによる動作の違いは、特に考慮しない

1. **アプリケーション概要**

2.1　アプリケーション名

― SpotMusic

2.2　アプリケーションコンセプト

― テーマ：音楽のストリーミングとユーザー同士のコミュニケーション

― 詳細：Spotifyというアプリになるべく近づけ、さらにユーザー同士のコミュニケーション機能を実装した音楽アプリ。

― GitHub: <https://github.com/MasayaTakubo/SpotMusic>

2.3　機能一覧

・ログインしていない状態でできること

― アカウントの登録

・ログイン状態でのみできること

― メイン画面の表示

1. 音楽の再生
2. 音楽の検索
3. プレイリストの操作(作成、削除)
4. プレイリストの曲の操作(追加、削除)
5. アーティストのフォロー/フォロー解除
6. フレンド機能の操作
7. フレンド間の個人チャット
8. フレンドのプレイリストへのコメント
9. ログアウト

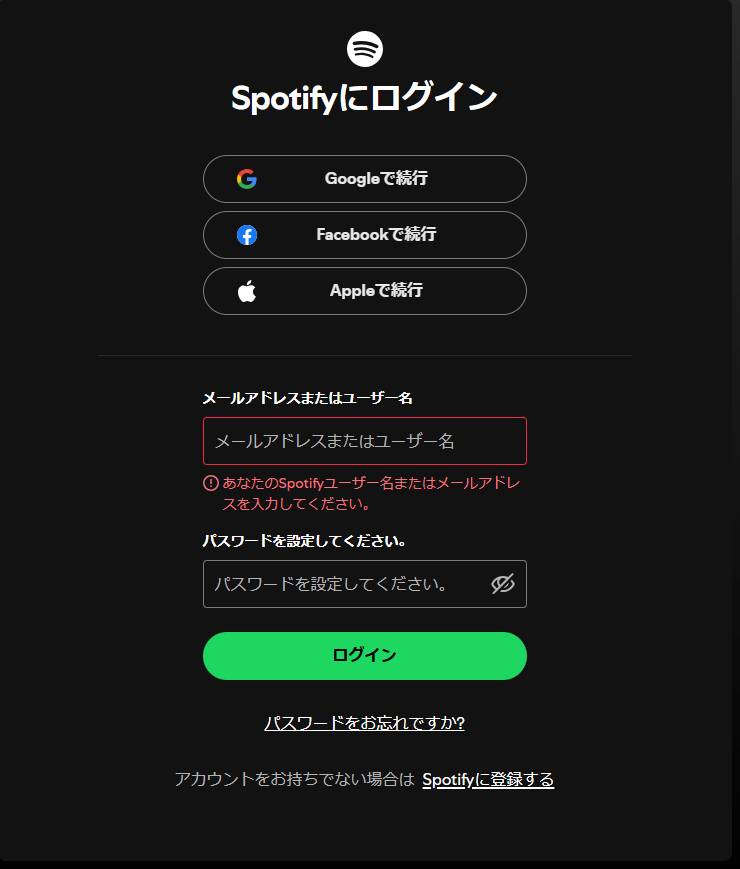
2.4　機能説明

― **ログイン画面**



ログイン画面にあるログインするボタンを押すと、Spotifyのログイン画面に遷移。

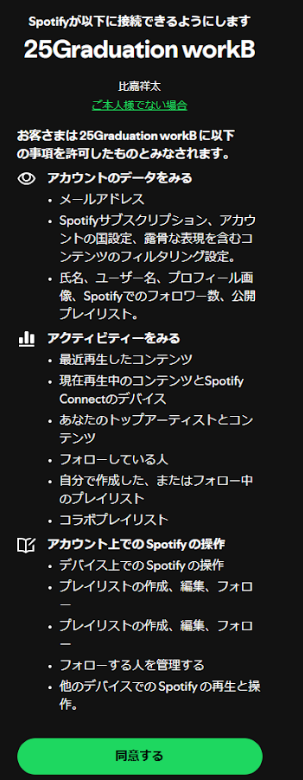
本アプリ(SpotMusic)の利用にはSpotifyのアカウントが必要。



SpotifyAPIを用いて、Spotifyに登録されているアカウント情報をもとにOAuth認証を行う。

ユーザー名またはメールアドレスと、パスワードを入力し、ログインボタンを押すと

SpotifyAPI認証画面へ遷移。



同意を押すと、SpotifyAPIの利用準備が整いメイン画面へ遷移する。

一度認証すれば、その後は表示されない。

アカウントを登録していない場合は、サインアップボタンからSpotify公式サイトへ遷移し、登録できるようになっている。

― **アカウント登録画面**

サインアップボタンから新規登録画面に進む



全ての情報を入力し、登録をする。

登録完了すると、Spotifyのアカウントを用いて、SpotMusicの利用が出来るようになる。

― **メイン画面**



左側に自身のプレイリスト、中央にフォロー中のアーティスト、最近再生された曲、

最も再生したアーティスト一覧、最近リリースされた新着アルバムが表示される。

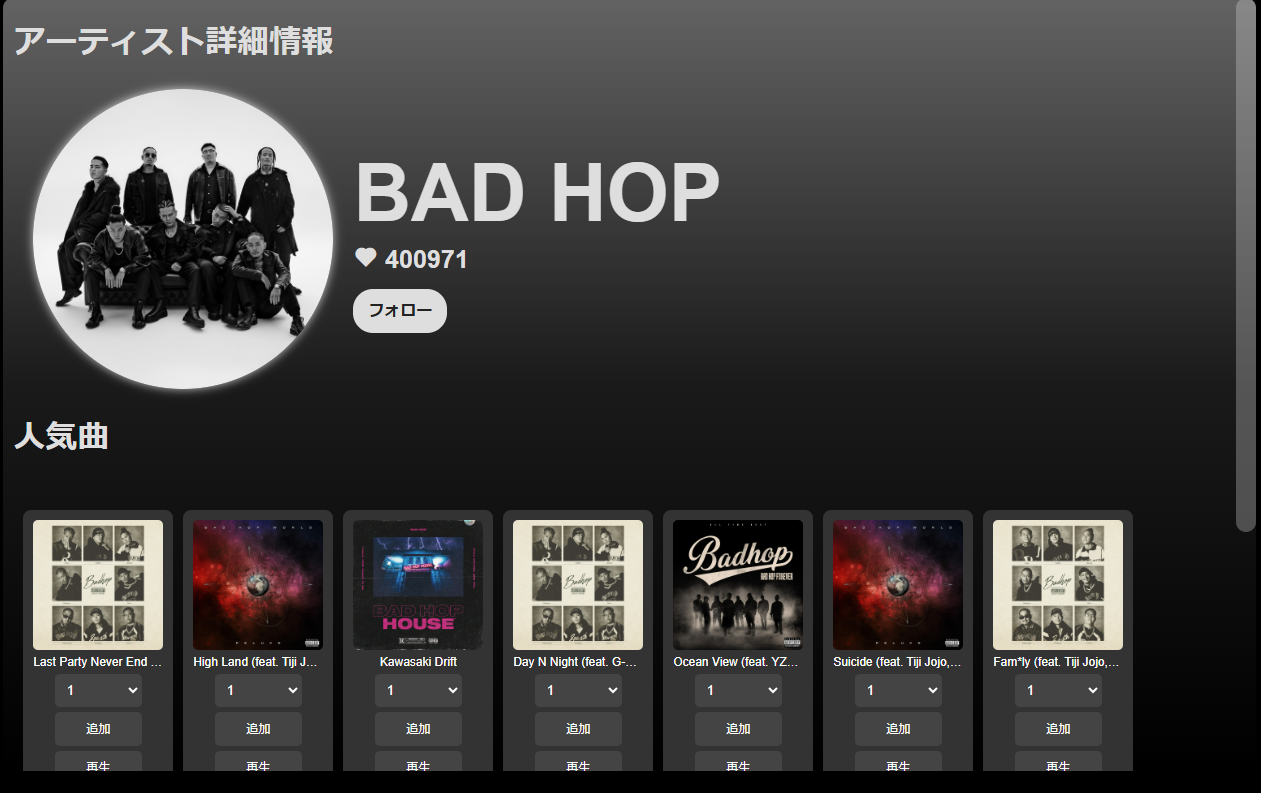
曲の再生中は右側に再生している曲の情報が表示される。

これらの情報は全て、Spotify側で保持している情報と連動している。

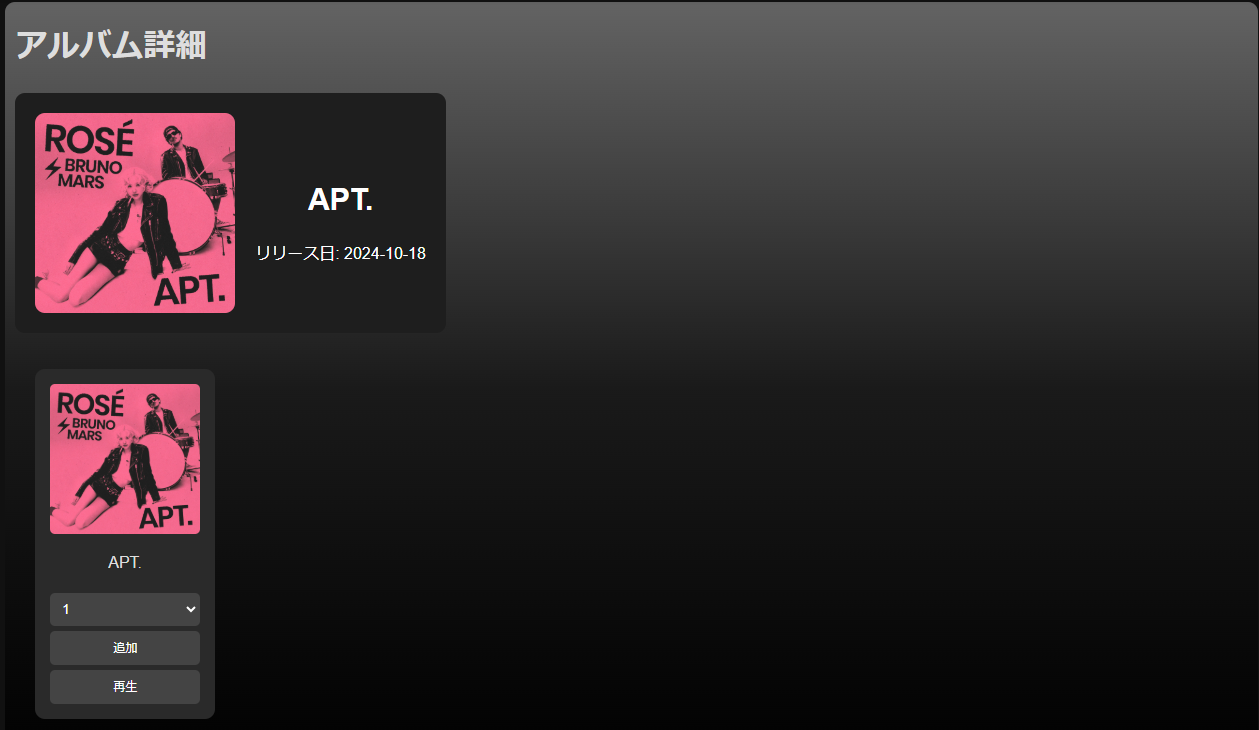
後述する各機能の操作も、SpotMusic側だけでなく、Spotify側にも反映される。

アカウント作成後間もない状態だと、表示できる情報が無い為新着アルバムのみ表示される。

― **アーティスト詳細ページ**



― **アルバム詳細ページ**



中央に表示されているアーティストや新作アルバムをクリックすると、詳細ページへ遷移。

― **リロードボタン**



左上のリロードアイコンを押すと、最新の情報に更新される。

― **ホームボタン**



各ページに戻るボタンが無い代わりに、検索バー左のホームボタンを押す事でトップページへ切り替わる。

― **アカウントアイコン**



アカウントアイコン。ログイン中のユーザー名が表示されている。

ボタンを押すと、アカウントの詳細情報(Spotify登録時のメールアドレス等の個人情報)を確認できるSpotifyのページに遷移する。

フレンドリストボタンを押すと、ユーザーのフレンドリストページに遷移。

ログアウトボタンを押すと、ログアウトが出来る。

― **プレイリスト操作**





(＋)ボタンを押し、任意のプレイリスト名を入力後、作成ボタンを押すとプレイリストが作成できる。



プレイリスト右側の(ー)ボタンを押すと削除が行える。

各プレイリストをクリックすると、プレイリストの詳細ページに遷移し、曲の再生が行える。

― **曲の検索**



任意のキーワードを入れ🔍ボタンを押すと、Spotifyが保持しているデータの中からキーワードがヒットする曲、アーティスト、プレイリストが検索できる。



デフォルトではすべての結果が表示されているが、タブを押すことによりジャンルごとのソートが可能。

各コンテンツをクリックすることで、其々の詳細ページへ遷移。



タブから任意のプレイリストを選択し、追加ボタンを押す事でプレイリストへ曲の追加が可能。

― **フォロー操作**



アーティストの詳細ページにあるフォローボタンを押すと、アーティストをフォローできる。

フォロー中のアーティストは、フォロー解除ボタンを押す事でいつでもフォロー解除可能。

フォローを行う事で、ログイン時トップページにアーティストが表示されるのでお気に入りのアーティストページにアクセスしやすくなる。

― **再生プレイヤー**



常に画面下部に、プレイヤーが表示されている。

各ページの再生ボタンを押すことで曲が再生される。

プレイヤーは”再生”、”停止”、”前の曲”、”次の曲”、”プレイリストリピート”、”1曲リピート”、”シャッフル”の操作ができる。

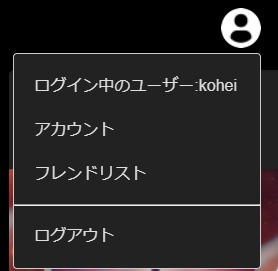
この再生プレイヤーは、Spotifyが提供している Web PlayBack SDK を使用している。

SDKを通してSpotifyの再生デバイスが作成され、SpotMusic上での楽曲の制御が行えるようになっている。

SpotMusicを使用する際、Spotifyは非アクティブでも問題は無い。

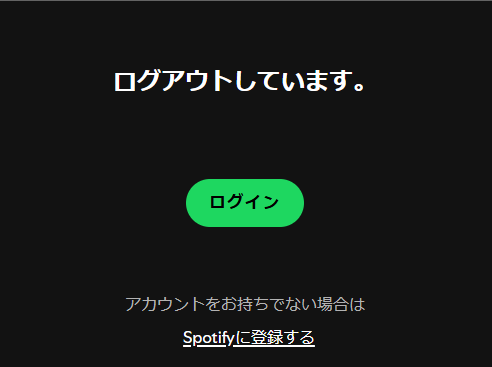


― **ログアウト画面**



右上のユーザーアイコンをクリックし、ログアウトボタンを押す。

別ウィンドウが開き、Spotify側と、SpotMusic側双方のログアウトが完了し、ログイン画面へ遷移する。(別アカウントの使用には双方のログアウトが必須の為)



― **フレンドリスト画面**

右上のユーザーアイコンをクリックし、フレンドリストボタンを押す。

この画面より、ログインユーザーが他ユーザーに対するアクション、ステータス、他ユーザーのプレイリストを確認することができる。

また、ユーザーリスト、ブロックリスト、チャット画面、プレイリストへのリンクが存在する。

アクションとはフレンドへに対するアクション（フレンド申請の許可、拒否、取消、ブロック）

ステータスとはフレンドへの申請状況（申請中、キャンセル）を意味する。



― **ユーザーリスト画面**

SpotMusicにアクセスしたことのあるユーザーのみ表示される。

この画面から対象ユーザーに対するフレンド申請、申請の解除、ブロック、ブロックの解除ができる。

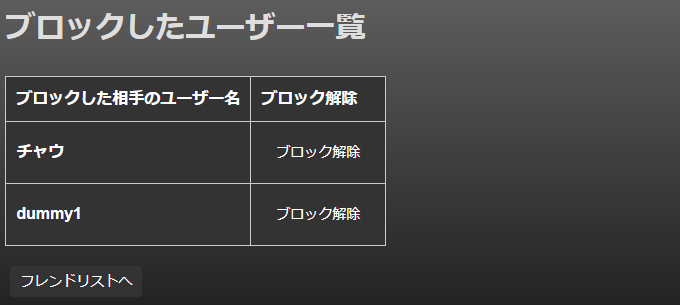
また、他ユーザーに対する申請状況の確認をすることができる。



― **ブロックリスト画面**

この画面からログインユーザーがブロックしたユーザーの一覧を確認することができる。

また、ブロックの解除をすることができる。



― **チャット画面**

フレンドリスト画面より、ログインボタンをクリックすると別タブでチャット画面が起動する。

相互のフレンド関係があることがアクセスができる前提条件となる。

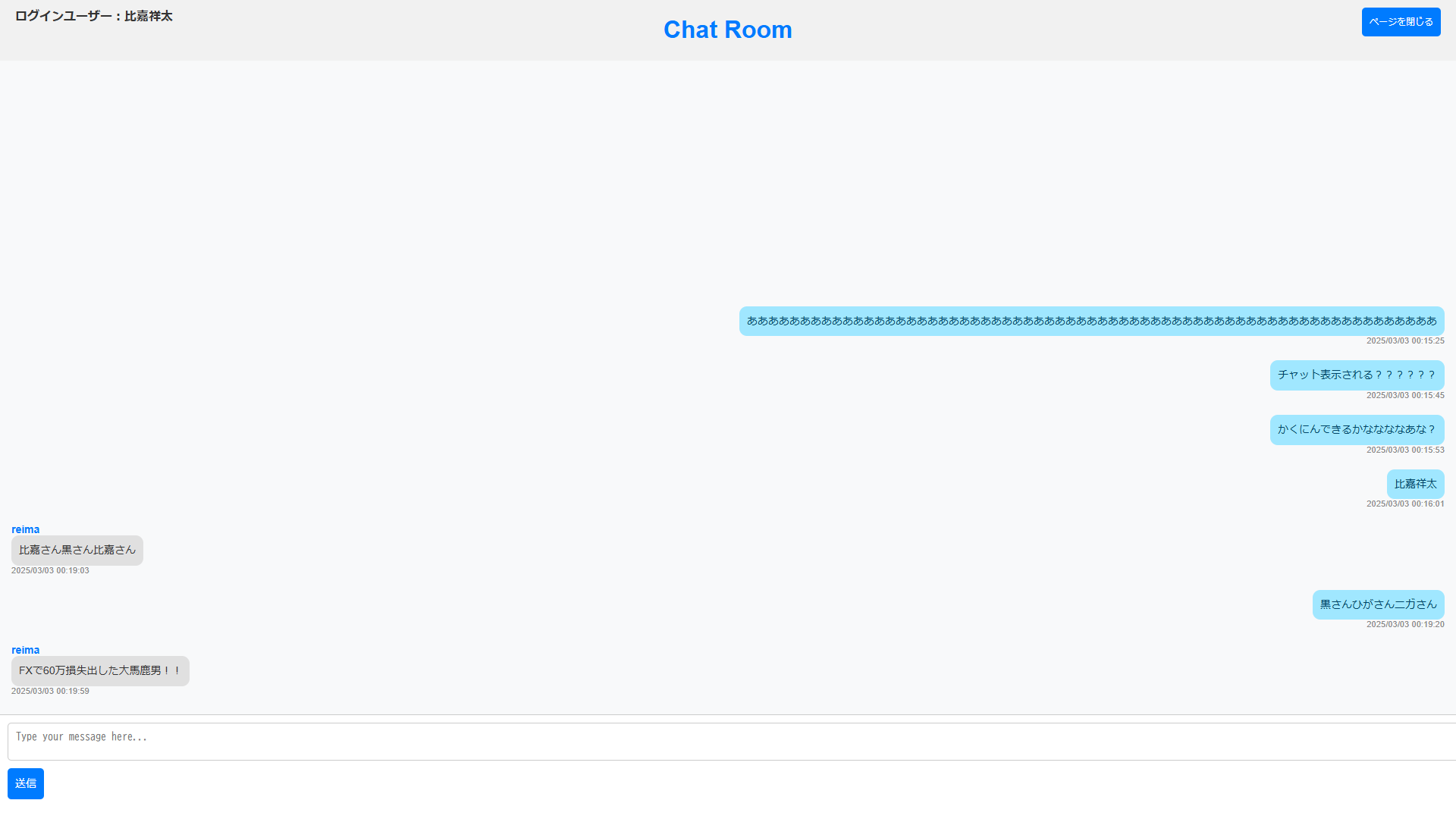
相互フレンドの場合は機能制限無し。

相手をブロックしている場合は、メッセージを送信することができず、受信することもできない。

相手にブロックされている場合はメッセージを送信することはできるが相手方は受信することができない。（ブロックされていることに気づかない仕様）

クロスサイトスクリプティング、SQLインジェクションを対策。タグ文字はエスケープされるようになっている。

送信したデータは即時画面に反映され、タブ間の非同期通信を実現している。



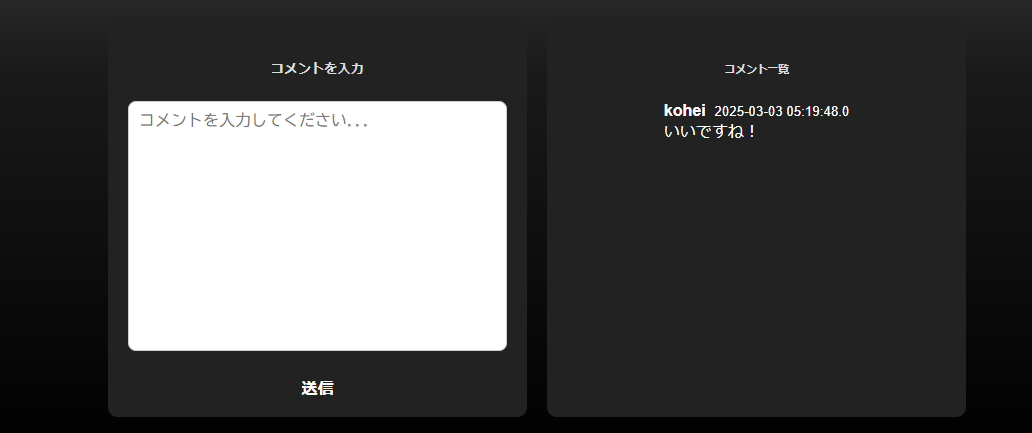
― **プレイリスト画面（コメント）**

フレンドリスト画面や検索などから、対象のプレイリストにアクセス。

フレンド関係のあるプレイリストへ複数ユ－ザーでコメントすることができる。

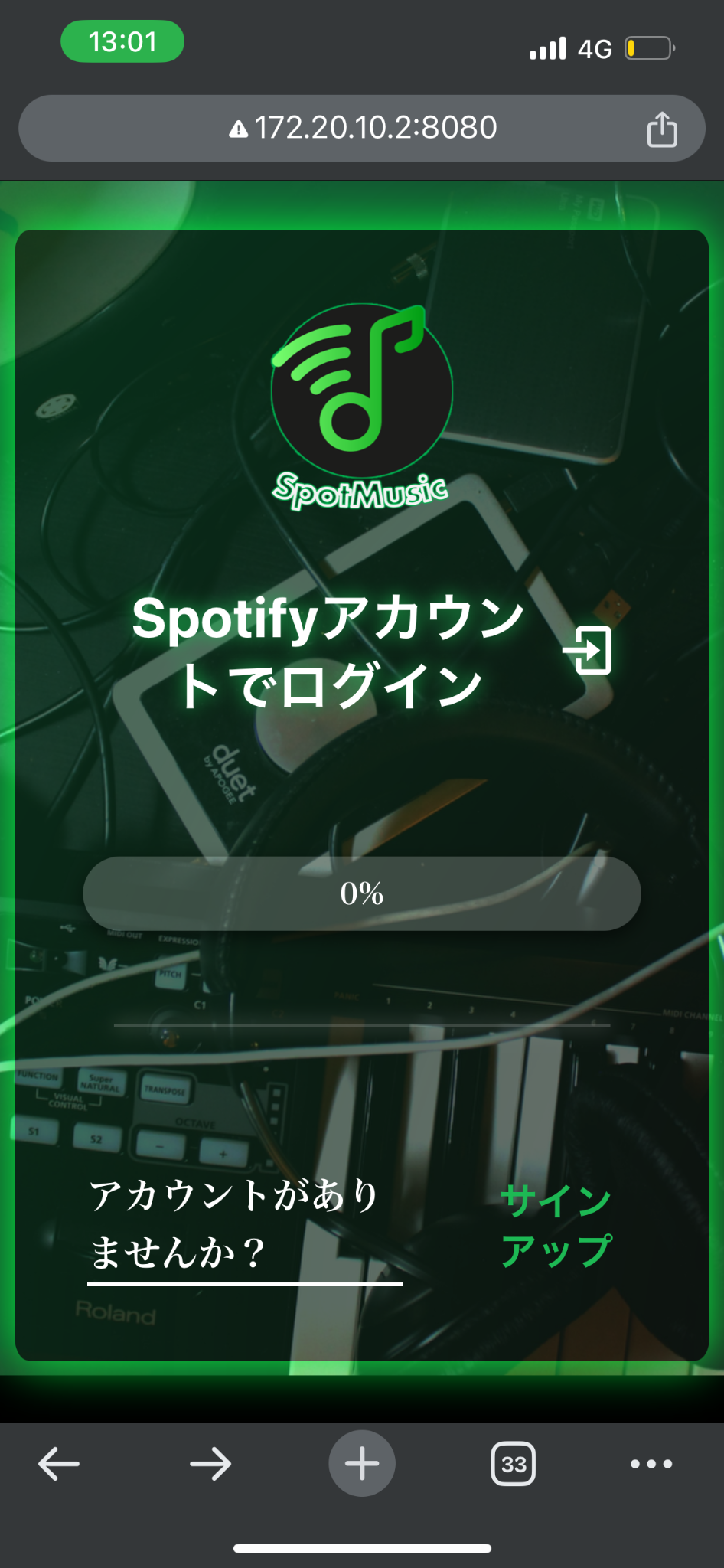
コメント一覧には、ユーザー名、投稿時間、コメント内容が、最新順で表示されている。

クロスサイトスクリプティング、SQLインジェクションを対策。タグ文字はエスケープされるようになっている。



― **モバイルデバイス（ログイン画面）**

モバイルデバイスでアクセスした場合、画面サイズが自動で適したサイズに切り替わる。

― **モバイルデバイス（フレンド画面）**

モバイルデバイスからのアクセスの場合、独自機能のみ利用できる制限となっている。

自動的にフレンドリストページへと遷移する。

― **モバイルデバイス（ユーザーリスト画面）**

****

― **モバイルデバイス（ブロックリスト画面）**

****

― **モバイルデバイス（チャット画面）**

****

**参考サイト等**

4.1 参考サイト

― Spotify

<https://open.spotify.com/intl-ja>

　　4.2　参考文献

― ChatGPT

<https://chat.openai.com/>

― Qiita

<https://qiita.com/>

― SpotifyforDevelopers

<https://developer.spotify.com/documentation/web-api>

― GitHub / spotify-web-api-java

<https://github.com/spotify-web-api-java/spotify-web-api-java>

― 授業資料

**制作を終えた感想**

**・起こった不具合**

― APIのリクエスト過多によるアクセス制限

― すべてが1ページで表示するためCSS修正が難しい

― 取得した情報が全て英語表記になる。

― データが取得できているのに画面に表示が出来なかった

― シームレスな画面遷移を実現する際のデータの適切な取得

― 開発環境の違いが原因で実行できなかった

― コメント送信後の画面遷移

**・良かった点**

― 難しい問題をチームで共有でき、グループワークをしている実感がした。

― 一緒に協力して、効率よく作業が進んだ

― 何かを実装するときに、調べる力がかなり上がった。

― 報連相などグループワークで必要とされることを十分行うことが出来た。

― これまでの技術の応用。APIの活用ができた。

― API のレスポンスデータを適切に解析する。

― 早い段階で役割を決めたこと

**・苦労した点**

― 言葉の壁

― チームワーク

― 再生プレイヤーの実装

― チャット画面の非同期通信による表示の更新

― 過去最多メンバーでの作業分担

― 様々なアプリケーションの使い方

　　**・改善点**

― APIのエンドポイントの利用可否まで調査が行き届かなくて時間ロスが目立った

― もっとコミュニケーションを強化する必要があった。

― 今回実装しなかったSpotifyにある機能の実装(歌詞とか)

― 計画の段階で決めるべきものを決め切らなかったこと

― キャッシュを利用したレスポンスタイムの短縮

― 重要な資料はタグ付けやカテゴリ分けで、後で簡単に参照できます

― コメント送信後の画面遷移をもっとスムーズに

**・個人コメント**

小池 : SpotifyAPIを用いたデータの取得について担当しましたが、作成途中で取得できない情報もある事がが判明して、路線変更した経緯もありとても忙しかったです。チームのメンバーと話し合いながら作成を進めることができて、とてもやりがいを感じました。

タンチャウ : 画面遷移無しで表示させるWEBアプリの制作だった為、CSSの編集が非常に難しかった。ある部分を修正したいとき、他の部分に影響を与えないように気を付けなければならず、意外と時間がかかってしまいました。

チームのサポートを受けたおかげで、完成させることが出来て良かったです。

齊野:再生プレイヤーの実装の際、SpotifyAPIと別で、SpotifyWebPlayBackSDKという再生用のAPIを併用しなければいけなくて実装が大変だったが、完成できて良かった。

佐藤 : チャット機能を担当したが、ユーザーが使いやすいようなレイアウトを考えたり、チャット機能に必要な情報の洗い出し、非同期通信の実現などLINEのようなチャット機能を実現する為には意外と多くの作業が必要で、度々問題に直面する事もあり苦労しました。

田久保 :操作画面をシームレスに利用できるように作成したが、必要なデータの取得方法など詰まる場面が多かったです。APIを利用し実際にリリースされているアプリのようなものを作成でき、達成感がとても生まれました。

横本 : APIを用いて実現出来ることと出来ないこと、誰が何の作業を担当するのかの割り振りや確認など作業前の準備が大事なんだと製作を通じて学べました。

張: 設計段階で各機能の関係が複雑で、上手く構成できない問題に直面したが、チーム内で話し合い、サポートを貰って解決できた。UMLの作成時にも同じ問題に直面したが、チーム内で協力する事により解決することが出来ました。

この製作での経験を通じて、チームワークと積極的なコミュニケーションの重要性を実感したので、今後に生かしていきたいです。