

# 研究室用グループウェアアプリの開発

情報科学科 27020731 細井 大輝

## 1 序論

西谷研究室では情報共有ツールとして Discord[1] を使用している。Discord は「オンラインゲームをしながら、世界中の友達とコミュニケーションをとる」[2] という課題を解決するために開発されたという背景がある。しかし、Discord 上での情報共有のデメリットとして以下の 3 点が挙げられる。

1. 図 1 のようにデータは降順に表示されるため、過去に共有されたデータはスクロールして遡るしかない。
2. データのタグ付け機能が存在しないため、データの絞り込みが困難。
3. 日付ごとのデータ管理ができないため、整理しづらい。

結果として共有されたデータが整理されずグループメンバーが混乱してしまい、継続的に利用されなくなる。そこで、データを参照する際の混乱を避け、データを共有・見返す作業を癖付けることを目的としたグループウェアアプリの開発を行う。

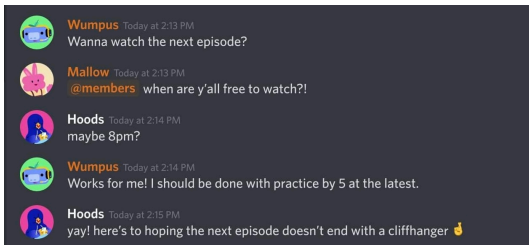


図 1 Discord で情報共有した場合の画面。

## 2 開発手法

開発環境として Ruby on Rails(Rails)[3] を選定した。Rails の利点は以下の 4 点である。

1. 広いプラットフォームからのアプリ利用が可能。
2. 日本語のドキュメントが充実しているため開発ハードルが低い [4]。
3. ライブラリが豊富。
4. クラウドプラットフォーム Heroku[5] との連携が容易。

データの分類方法としては「超」整理法 [6] の時間軸をキーとして分類する方式を採用する。Rails と「超」整理法の時間軸整理を組み合わせ、データを日付ごとに管理することによって出席状況や、その日に取ったメモなどを一目で把握できる Web アプリを開発する。

## 3 結果

今回、開発した Web アプリでの情報共有のメリットは以下の 2 点である。

1. 投稿されたデータは保存したい日付を 1 つのカラムに保持しているため、日付ごとのデータ管理が容易。
2. 投稿されたデータ一覧画面では、タグでのデータ絞り込み機能が利用可能。

また、どちらのアプリもログイン済みであることを前提として、投稿されたデータを参照するまでのアクション数を比較する。Discord ではサーバーの選択、チャンネルの選択、画面のスクロールなど最低でも 3 つのアクションを要する。私が開発した Web アプリでは日付選択の 1 アクションでデータを参照することが可能である。



図 2 参照したい日付に保存されたデータを参照する一連の動作。

## 4 今後の課題

今回は基本的な Create(生成), Read(読み取り), Update(更新), Delete(削除) 処理 (CRUD 処理) に加えて、データの絞り込み機能、グループ作成、メンバー招待・参加機能を開発した。今後は共有データの見逃しを防ぐための通知機能、ローカル PC 上で作成したメモからこの Web アプリへの送信スクリプトを開発する。また、閲覧回数などの要素から共有データを重みづけし、不要なデータは自動的に削除する機能も組み込み、更なる共有データの整理、Web アプリのパフォーマンス向上を図る。

## 参考文献

- [1] Discord, <https://discord.com>.
- [2] Discord 会社概要, <https://discord.com/company>.
- [3] Ruby on Rails チュートリアル, <https://railstutorial.jp>.
- [4] Ruby on Rails ドキュメント (v7.0.0), <https://railsdoc.com>.
- [5] WHAT IS HEROKU?, <https://www.heroku.com/what>.
- [6] 野口 悠紀雄, 「超」整理法—情報検索と発想の新システム (中公新書), 中央公論新社, (1993).