

# 企画書

チームしのと

2023 年 1 月 13 日

## 1 目的

このプログラムは普段から Discord をコミュニケーションツールとして利用している大学生をターゲットとし、自己管理に役に立つツールを提供する。仲の良いグループで集まっている Discord グループにて、履修科目などのチャンネルの作成や、学年・研究室などのロールの設定についての自動化を主な機能とする。

## 2 構想

今回、Discord.js を利用して Discord の Bot を作成する。

従来の履修・課題管理アプリケーションでは、一人で使うことが前提とされているため、友達をコミュニケーションをとるには別のアプリケーションを使う必要があった。それを解決するために Discord という一つのアプリケーションで履修・課題管理アプリケーションの模倣を試みた。今までは Discord で履修する科目の課題を仲の良い友達同士で共有するとき、一つのチャンネルで履修する科目を共有すると話が複雑になってしまう。複雑になってしまうというのは、一人の方がある科目に対して話題を振ったとしても、直ぐに誰かが返信をすることができないと別の方が他の科目のことを話せなくなり、また他の課題を途中で振ると、複数の科目の話題が混じってしまう。なので、サーバーの管理者がカテゴリ、チャンネルを科目ごとに作成する必要があった。Discord にはロールという機能があり、ロールが割り当てられていると特定のチャンネルを閲覧することができた。ロールを作成せずに全科目を全員が見ることもできるが、その科目を受けない人は各々通知を消す必要があった。なので管理者が全員がそれぞれ履修する科目を作成して、その科目を受ける人が履修するならその人にロールを設定しなければならない。その手間と言ったら仲の良い友達が 5, 6 人だとして履修する科目が全員の合計で 10 個だとすると単純計算で 50 回くらいの動作を行うことになる。

それを改善するために、Discord 内のロール機能を利用し、ボタンを押すだけで自動的に履修する科目としない科目で振り分けることができる機能があると楽にチャンネルを管理できるようになると考えた。また、リマインダー機能を作成された科目チャンネルで行い、あえて全員が見れるようにすることで可視化され、友達同士で声を掛け合うことができる。そうすることで、「今週はいいや」等の気持ちに負けることが少なくなると考えた。

また、Bot として実装するため、さまざまなサーバーに同時に導入することができる。

以下、機能を記述する。

## 2.1 科目チャンネル作成

指定のコマンドと科目名を誰かが入力することで、科目チャンネルとロールが生成される。コマンドから帰ってくる Bot のメッセージ内のボタンを押すことで、その科目のロールが割り当てられ、自分のチャンネル一覧に表示されるようになる。

各ロールが付与されることで他のユーザーからどの科目を履修しているか分かりやすくする。その人の履修科目が、メンバー一覧のから一目で確認でき、質問などがしやすくなる。

## 2.2 ToDo リスト・リマインダー

各科目チャンネルを跨いだ ToDo を自分で確認することができる。リマインドは各科目チャンネルで行われる。リマインドの通知にアクションを入れることで ToDo から消える。



図 1 実際の操作画面



図 2 実行結果

### 3 作品の独創的な点

現在存在している ToDo Bot では汎用性は高いが、大学生にとってもっと便利な Bot を作成することができると考えた。また、大学生向けのソフトウェアの機能やその他の機能を追加することで一つの Bot で完結することができる。

Discord というアプリケーションでは、テキストチャットとボイスチャットともに機能が充実しており、そこで利用できるようにすることで複数のアプリケーションを跨ぐことなくコミュニケーションをとることができる。加えて、日常的に利用するアプリケーションなので、頻繁にメッセージを確認する。それによって、端末の通知を切っていたとしてもリマインダーとしての機能を強めることができる事が Bot として作成する強みだと考えた。

Slack を選ばなかった理由としては、Slack は大学や企業で業務的な使い方をするのが一般的であり、Discord は Slack よりも普段の生活に浸透しているためである。また、機能の一つであるロールの生成に関して、Slack にはない機能であり、履修関連の情報表示として実用的であると考えた。そして、通常状態で自由に通話を開くことができない。これらより、Discord を開発対象とした。

### 4 プログラムの詳細

JavaScript で Bot を記述し、自宅内のサーバーからデプロイする。また、Discord.js というモジュールを用いて、サーバーから Discord に接続する。プログラムは、index.js という Bot の具体的な挙動を示したファイルと、Discord へのアクセスのための APIKey が格納された.env ファイルがある。そのほかに、node.js のモジュールのための package.json や、node\_modules などがある。

ここで、自宅からデプロイするとあるが、それは予算的な理由による。24 時間稼働のサーバーであるため、無料枠では実現できず、自宅で簡易的なサーバーを構築した方が良いと考えたため、自宅からのデプロイとした。

## 5 制作計画

実装内容の役割分担は以下のようになっている。

表 1 役割分担

	担当者
科目チャンネル関連	内山
リマインダー関連	安齋
TODO リスト関連	青木

最初の半月で詳細な開発の方向性を話し合い全員でベースを作る。1 か月で各々の箇所を作成し，残りの時間でコードを合わせデバックを行う。実際に使ってみて必要だと思った機能があれば追加する。