

大規模学会向け AIシフト自動生成システム 要件定義書

作成日: 2025-12-10

更新日: 2025-12-10

対象イベント: 2025年度 学会(3/24～3/29)

規模: 学生スタッフ200～300名、6日間

実装フェーズ: Phase 1(スプレッドシート連携版)

1. システム概要

1.1 目的

本システムは、学会運営における学生スタッフのマッチングとシフト作成を自動化し、

- 人手によるシフト調整工数の大幅削減
- 欠員や発表時間とのバッティングを考慮した安定した配置
- スタッフのスキルや経験を最大限活かした配属

を実現することを目的としています。

Googleフォームで収集した応募情報をもとに、
AI+最適化ロジックにより「誰を・いつ・どの係に」配置するかを自動で算出し、
結果をGoogleスプレッドシートに出力します。

1.2 対象イベント

- 期間: 2025年3月24日(月)～3月29日(土)の6日間
- スタッフ募集数: 200～300名
- 実働スタッフ: 日・時間帯によって変動
- 勤務時間帯: 午前・午後・夕刻・(懇親会枠)などの時間帯区分で管理

1.3 解決したい課題

- 200～300名規模のスタッフに対し、
発表時間・スキル・性別・希望日程などを考慮しながら
手作業でシフトを組むのは非常に負荷が高い。

- 発表者や経験者を適切なポジションに配置したいが、人数が多く、全体最適での配置が難しい。
- 欠員や条件不足がどこに出ているかを、ひと目で把握しづらい。

→本システムにより、「応募データを投入すると、ほぼ完成形のシフト案が自動で出てくる状態」を目指します。

2. 実装フェーズと今回の範囲

2.1 フェーズ全体像

- **Phase 1(今回) :**
Googleフォーム＋スプレッドシート＋マッチングエンジンによる
「バックエンド中心の自動マッチング」
- **Phase 2(ご要望あれば) :**
管理者向けのWeb管理画面(ブラウザから実行・簡易調整)
- **Phase 3以降(ご要望あれば) :**
スタッフマイページ、メール通知、出欠管理、LINE連携などを含む
フルWebシステム

2.2 Phase 1(今回)の範囲

実装する内容

1. Google Apps Script による スプレッドシート構成の自動生成
2. Googleフォームからの応募データを格納する「応募者データ」シート設計
3. 係マスター・必要人数・マッチング設定シートの設計
4. 応募データ・マスタデータを読み込み、
最適な配置を計算するマッチングエンジン
5. 発表者の発表時間帯を自動的にシフトから除外
6. 結果をスプレッドシートに出力する仕組み
7. 管理者がコマンド1本でマッチングを実行できる運用

Phase 1では行わない(将来拡張)

- ブラウザ上の管理画面(Web UI)
 - スタッフ向けマイページ
 - 自動メール通知機能
 - ドラッグ & ドロップでの視覚的なシフト編集UI
 - AppSheetを使った出欠打刻アプリ
(※スプレッドシート構成は、将来のAppSheet連携を見据えたものにします)
-

3. Phase 1 技術構成と運用イメージ

3.1 使用するサービス・技術

コンポーネント	内容
Googleフォーム	学生スタッフの応募受付
Googleスプレッドシート	応募データ・マスタ・結果の一元管理
Google Apps Script	シート構成の自動作成、初期設定
Google Sheets API	マッチングエンジンとスプレッドシートの連携
Python + 最適化ライブラリ	マッチングロジックの実装(運営側PCまたはサーバ上で実行)

※ クライアント様側で特別なサーバ構築は不要で、
基本的には「担当者PCから実行するバッチツール」というイメージです。

3.2 運用フロー(Phase 1)

1. 応募開始前
 - Googleフォームを準備(質問項目は本書付録をベースに設計)
 - スpreadsheetとApps Scriptをセットアップ
 - 「係マスター」「必要人数」「マッチング設定」を入力

2. 応募期間中

- 学生がGoogleフォームから応募
- データは自動的に「応募者データ」シートへ蓄積

3. 応募締切後(マッチング実行)

- 管理者がマッチング用スクリプトを実行
- スクリプトが以下を自動で実施:
 - 応募データの読み込み
 - 発表時間帯の除外
 - 条件を満たす最適な配置の計算
 - 結果を「マッチング結果(スタッフ別／時間帯別)」シートに書き込み

4. 結果確認・微調整

- 管理者がスプレッドシート上で結果を確認
- 必要に応じて、一部手動で修正

4. イベント構成(日時・時間帯・係)

4.1 日程と時間帯

対象期間:2025年3月24日(月)～3月29日(土)

時間帯は、運営の実態に合わせて次のように区分します:

日付	時間帯区分	時間	備考
3/24(月)	終日	8:00～17:00	準備日(設営中心)
3/25(火)	午前	8:00～13:00	準備日
	午後(業務オリン)	13:00～17:00	全スタッフ参加必須
3/26(水)	午前	8:00～12:00	会期1日目

	午後	12:00～17:00	
	夕刻	17:00～20:00	
3/27(木)	午前	8:00～12:00	会期2日目
	午後	12:00～17:00	
	夕刻	17:00～20:00	
	夜(懇親会)	17:00～20:30	懇親会係専用枠
3/28(金)	午前	8:00～12:00	会期3日目
	午後	12:00～17:00	
	夕刻	17:00～20:00	
3/29(土)	午前	8:00～12:00	会期4日目
	午後	12:00～17:00	

- 勤務可能時間帯は、フォーム上では 複数選択式 とします。
- 午前のみ・午後ののみなどの「部分的な勤務」も受付可能です。

4.2 係の種類と役割(13種類)

ID	係名	固定性	主な役割・条件
01	年会本部	全日固定	運営中枢。PC・Zoom上級、全日程参加必須
02	総合案内	時間帯替わりOK	案内対応。英語力あれば優先
03	参加証再発行	時間帯替わりOK	PC入力中心。正確性重視
04	クローケ	時間帯替わりOK	荷物の受け渡し。体力・スピード重視

05	会場係	時間帯替わりOK	会場案内・進行補助。Zoom中級、英語あれば優先
06	会場チェック係	時間帯替わりOK	配信チェック・機材確認。Zoom・PC中級、イベント経験者優先
07	試写室係	時間帯替わりOK	発表データ確認など、静かな作業
08	機動運用係	時間帯替わりOK	欠員・トラブル時の応援要員(会場係・会場チェック両方の素養)
09	設営係	兼務可能	会場設営。男性優先(力仕事)
10	展示係	時間帯替わりOK	企業・展示ブース対応。英語できると尚可
11	ポスター係	時間帯替わりOK	ポスター会場案内・誘導
12	懇親会クローケ	懇親会専用	懇親会時間に参加できるスタッフ
13	懇親会受付	懇親会専用	懇親会受付対応

重要ポイント

- 業務オリエンテーション(3/25 午後)は全員参加必須
→ この時間帯に参加できない学生は、「不採用」として扱います(マッチングロジックで自動除外)。
 - 年会本部は イベント期間を通して同じメンバーを想定しています。
-

5. データ管理(スプレッドシート構成)

Phase 1 では、すべてのデータを Google スプレッドシートで管理します。

5.1 シート一覧

1. 応募者データ

- Googleフォームから自動反映される生データ

2. 係マスター

- 各係の条件・優先度を定義

3. 必要人数

- 日付 × 時間帯 × 係ごとの必要人数を設定

4. マッチング設定

- スコアリングの重み(スキル・全日程参加など)

5. マッチング結果_スタッフ別

- 各スタッフが「どの時間帯・どの係」に配置されたか

6. マッチング結果_時間帯別

- 各時間帯・係に「誰が何人」入っているか

5.2 応募者データ(主な項目)

- 基本情報: 氏名、ふりがな、メールアドレス、学校名、学年、電話番号、住所 等
- 勤務可能時間帯(15枠):
 - 例: 3/26 午前 / 3/26 午後 / 3/26 夕刻 ...
- 学会発表:
 - 発表予定の有無
 - 発表日(最大3件)
 - 発表時間帯(午前・午後・夕刻)
- スキル:
 - PCスキル(初級～上級)
 - Zoomスキル(初級～上級)
 - 英会話(日常会話レベル / なし)

- イベントバイト経験の有無・内容
- 備考欄(領域の希望・配慮事項など)

5.3 必要人数シート

- 行:係(年会本部・会場係など)
- 列:日付×時間帯(例:3/27 午前 / 3/27 午後 / 3/27 懇親会)
- セルの値:必要人数(例:会場係 3/26 午前 = 50名)

→ 入力いただくのは、主にこの「必要人数」と「係マスター」の内容です。

5.4 マッチング結果の出力イメージ

スタッフ別シートの例

- 行:スタッフ1人ずつ
- 列:
 - 氏名、ステータス(採用／補欠／不採用)
 - 各時間帯に割り当てられた係名(例:3/26 午前 = 会場係)
 - 総勤務時間帯数、適性スコア、備考(発表時間除外など)

時間帯別シートの例

- 行:日付 × 時間帯 × 係
- 列:
 - 必要人数／配置人数／達成率
 - 配置されたスタッフ一覧
 - 不足・過剰に関する警告メッセージ

6. マッチング機能要件

6.1 採用ロジック(誰を採用するか)

1. 業務オリエン参加チェック

- 3/25(火)午後(業務オリエン)に参加「不可」のスタッフは自動的に不採用とします。

2. 適性スコアの算出

- 以下の要素を点数化し、合計点で評価します。
 - 参加可能な時間帯の多さ
 - PC・Zoom・英会話などのスキルレベル
 - イベントアルバイト経験
 - 全日程参加可かどうか

3. 採用・補欠・不採用の区分

- 総必要人数 × 採用倍率(例:1.2倍)を採用候補数とし、
- 上位を「採用」、続く一定数を「補欠」、残りを「不採用」として区分します。

6.2 配属ロジック(どこに配置するか)

最適化ロジックにより、次の条件を満たしながら配置を行います。

必須制約

- 各時間帯・各係の 必要人数をできる限り満たす
- 業務オリエン(3/25 午後)は 全スタッフ参加
- 年会本部は 全日・全時間帯を通じて固定メンバー
- 1人のスタッフは 同じ時間帯に1つの係まで
- 応募者が「勤務不可」とした時間帯には配置しない
- 係の最低限のスキル条件を満たさない場合、その係には配置しない
- 発表者は 発表時間帯に配置しない

優先される考え方(目的)

- スキルが高い人を、スキルが必要な係へ優先配置
- イベント経験者を、会場チェック係などの重要ポジションに複数時間帯配置
- 設営係には男性比率を高めに設定(力仕事を想定)
- 「全日程参加できる人」を高く評価し、安定した体制を整える

6.3 特殊な配属ルール

- 設営係(3/24 終日)
 - 設営に参加可能な学生から優先的に配置
 - 他の日程では別の係と兼務可能
- 撤去／最終日の重労働枠(3/29 午後など)
 - 男性を優先しつつ、必要人数を満たすよう配置
- 懇親会係(3/27 懇親会時間帯)
 - 該当時間帯を「可」と回答した学生のみを対象
 - 会場係などとのバランスを見ながら配置
- 発表者
 - 発表日・発表時間帯は、勤務時間から自動的に除外
 - 発表のない時間帯については、他スタッフと同様に係に配置

6.4 出力内容

マッチング実行後、以下が自動的に生成されます。

- 採用／補欠／不採用の区分
- 各スタッフの「時間帯ごとの係配置」
- 各時間帯の「必要人数／配置人数／不足・過多の有無」
- 必要人数が満たせなかった枠の一覧(警告)

7. 非機能要件

7.1 性能

- スタッフ数:最大 300名
- 時間帯数:最大 15枠
- マッチング計算時間:
 - 目安:60~120秒以内(PCスペックに依存)

7.2 セキュリティ・権限

- 個人情報はクライアント様管理の Google スプレッドシート上で管理
- シートの共有権限は、必要な運営メンバーのみに限定
- マッチングエンジンはサービスアカウント経由でアクセスし、認証情報は安全に保管

7.3 運用面

- マッチング実行は、クライアント側担当者(または弊方)が行う想定
 - 結果確認・微調整は、スプレッドシート上で実施
 - 追加のシステム運用基盤(サーバ構築など)は不要
-

8. 制約事項

- 業務オリン(3/25 午後)参加不可の応募は、採用対象外となります。
- 本フェーズでは、
 - Web管理画面
 - メール自動送信
 - スタッフ向けマイページ
などは提供対象外です(スプレッドシートベースのバッチ運用となります)。

- キャンセル発生時の再調整は、
 - スプレッドシート上の修正
 - 必要に応じて再マッチング実行
という運用を想定します。
 - Google Sheets API の利用制限(呼び出し回数)により、
1度に扱えるデータ量や実行頻度には上限があります(通常規模では問題ない想定です)。
-

9. Phase 2以降の拡張イメージ

今回の Phase 1 を土台として、以下のような拡張が可能です。

9.1 Phase 2: 管理画面(管理者向けWeb UI)

- ブラウザ上から「マッチング実行」ボタンでシフト生成
- パラメータ(スキル重み・採用倍率など)のGUI調整
- 結果の一覧表示・簡易な手動修正
- CSV出力／印刷レイアウト

9.2 Phase 3: 出欠管理・通知連携

- AppSheetやQRコードを用いた出欠管理
- 出欠状況を元にした当日再マッチング
- 採用／シフト確定のメール・LINE通知

9.3 Phase 4: フルWebシステム

- 応募フォームを含む専用Webシステム
- 学生向けマイページ(シフト確認・希望変更)
- 係長向け画面(自係のスタッフ管理)
- 学会ごとのテンプレート管理・複数イベント対応

