



SESマッチングシステム V7Pro 機能説明書

案件と要員を、利益幅とスキル一致で高速にスコアリングし、AI (Claude API) で上位候補の精密分析
・未知スキルの辞書拡張まで支援する、SES営業向けツール。



データ取込

CSV/テキスト+メールサーバー直取得

PapaParseでOutlook CSV取込
IMAP/POP3はHTTPプロキシ経由で取得



即時自動分類

案件/要員/その他に仕分け

本文から単価 (万円) を抽出
重複排除 (本文ハッシュ等)



マッチング

利益幅+スキル一致の総合スコア

最低利益幅・最低技術一致数・最低スコアを設定
ビュー切替: スコア順/案件別



AI & 辞書

未知スキル検出→AIで解決→辞書追加

スキル辞書: 14カテゴリで正規化
上位候補をAI精密分析 (強み/不足/推奨単価)



利用シーン

SES営業の「探す・比べる・送る」を短縮

🔍 照合の高速化

案件⇔要員の候補をスコアで優先順位付け

✉️ 提案メールの省力化

選択結果から提案文を自動構成し送信

🛡️ 運用の安全性

ライセンス・期限管理/設定は
LocalStorage保存

ヒーロー図 (概念)

案件 ↔ 要員 ↔ AI



案件メール

単価・スキル要件を抽出

➡️ 自動分類・スコアリング



要員メール

希望単価・スキルを抽出



AI精密分析

強み/不足/推奨単価・未知スキル補完

📖 辞書拡張

未知語を追加し精度改善

📄 結果出力

Excel/CSV/メール送信

📁 ローカル辞書+設定 (LocalStorage)

🔑 Claude APIキー

Scope : 読込 → 分類 → マッチング → 出力

v6 Pro



システムアーキテクチャと技術スタック

ブラウザUI (HTML/CSS/JS) を中心に、メール受信/送信はHTTPプロキシ経由、AI分析はClaude APIへ接続する構成。
辞書・設定はローカルに保持し、案件⇔要員のマッチング処理はクライアント側で実行する。



フロントエンド
単一HTMLで動作 (ブラウザ)

HTML / CSS / JavaScript
フォント: Noto Sans JP / M PLUS 1 Code
UI: タブ・モーダル・進捗表示



データ入出力
CSV/Excel変換と結果出力

PapaParse: CSV/テキスト取込
SheetJS/xlsx: Excel書出し
CSV: マッチング結果エクスポート



セキュリティ/ライセンス
ロック画面 + 期限管理

15桁コード (XXXXXX-XXXXX-XXXXX)
Web Crypto API (SHA-256) で検証
設定/状態: LocalStorage



外部連携
AI・メールサーバー接続

AI: Claude API (APIキー保存)
受信: IMAP/POP3 (HTTPプロキシ)
送信: SMTP (HTTPプロキシ, STARTTLS/SSL)



プロキシが必要な理由

ブラウザはIMAP/POP3/SMTPへ直接接続できないため、HTTPエンドポイントとして中継する。(本スライドは実装HTMLの仕様記載に基づく)

受信プロキシ (例)

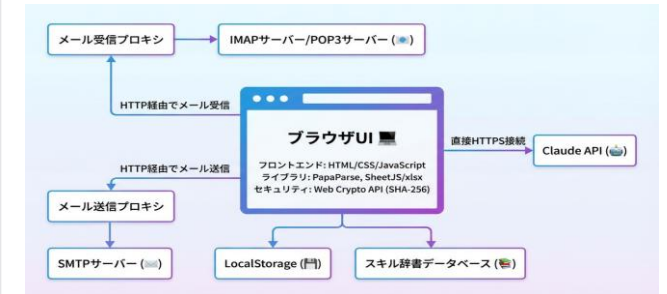
POST /api/mail → [{subject, body, from, date}]

送信プロキシ (例)

POST /api/send → {success:true}

構成図 (ブロック図)

Browser 중심



データ保持

辞書・設定は端末内保存

処理方式

分類・照合はブラウザ実行

Focus : 外部連携点 (メール/AI) とローカル保持



読み込みから分類までを一気通貫で自動化

本システムは、Outlook CSV/テキストまたは メールサーバー (IMAP/POP3) から取得したメッセージを 正規化・重複排除したうえで、案件/要員/その他へ即時分類する。さらに本文から単価 (万円) を抽出し、以降のマッチングに利用する。

① データ読込 (CSV/テキスト)

ドラッグ & ドロップで投入

PapaParseでCSVを解析
件名/本文/差出人/日付を抽出
形式差を吸収して内部モデルへ統一

② メール取得 (IMAP/POP3)

ブラウザ制約のためプロキシ必須

設定: ホスト/ポート/SSL/認証/フォルダ/件数
プロトコル: **IMAP / POP3**
取得: **HTTPプロキシ**経由でJSON化

③ 正規化・重複排除

精度と処理速度の土台

重複判定キー: 件名/差出人/本文
Web Crypto(SHA-256)でハッシュ化
同一メールの多重計上を防止

④ 自動分類 + 単価抽出

案件/要員/その他

キーワード/パターンで即時仕分け
本文から**万円**表記を解析し上下限推定
分類リストを単価順で表示 (案件:降順/要員:昇順)

受信プロキシ仕様 (実装上の前提)

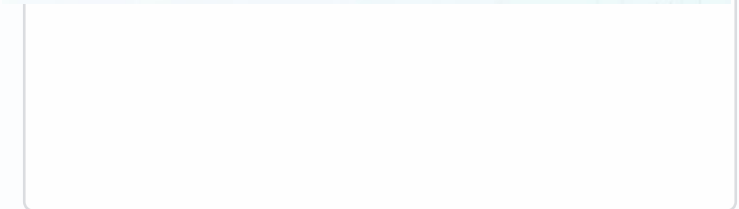
ブラウザからIMAP/POP3へ直接接続できないため、メール取得はHTTPエンドポイントとして中継する。返却は配列JSON (件名・本文・差出人・日付) を想定。

➔ Request
POST {proxy_url}
{host,port,ssl,user,pass,folder,limit,protocol}

➔ Response
[**{subject, body, from_name, from_addr, date}**]

処理フロー (概念図)

入力 → 正規化/重複排除 → 分類 → リスト表示



📊 統計表示
取込件数・分類件数・単価あり
等を集計

🔄 次工程
分類結果をもとにマッチング実行
等を集計

Outcome : 案件/要員リストが揃い、マッチング準備完了

v6 Pro

注: 分類ロジックはキーワード/パターンに基づく。単価は本文表記から推定し、以降のスコア計算に利用する。



条件フィルタと総合スコアで「勝ち筋の候補」を優先順位付け

案件×要員の全組合せから、除外ルールで候補を絞り、利益幅とスキル一致でスコアリングして並べ替える。UI設定（最低利益幅・最低技術一致数・最低スキルスコア）に準拠。



マッチング条件 (UI設定)
フィルタ強度を調整して精度を制御

最低利益幅 (万円)

案件単価 - 要員単価 が設定値未满是除外

最低技術一致数 (0/1/2/3)

技術カテゴリの一致が不足する候補を除外

最低スキルスコア (50-99)

しきい値を指定 (低いほど網羅)



スコアリング (概念式)
利益×スキルのバランスで順位付け

総合スコア =
利益幅 × 係数 + スキル一致点 + 技術一致加点

利益幅
案件-要員
(万円)

スキル
辞書正規化
後

技術
言語/FW等

並べ替え・ビュー

フラット: スコア順

案件別: 案件ごと
提案組立向き

絞込・全選択



除外ルール (既定)
提案可能性を担保するため、以下は自動除外。単価不明は設定で含める/除外を選択可能。

× 単価逆転
要員 > 案件

× 技術一致0
一致なし

× 利益不足
最低未滿

Tip: 設定値 (利益幅/技術一致/最低スコア) を上げるほど候補は絞られ、提案品質は上がる。

アルゴリズムフロー (概念図)

候補生成 → 条件フィルタ → スコア計算 → 並べ替え/表示

P4



実行体験
進捗バー・ステータス文
で処理状況を表示

次工程
上位候補をAI精密分
析へ連携

Outcome: 提案優先度が高い組み合わせを上位に集約

v6 Pro

注: 係数や配点は運用方針に応じて調整可能 (UIは利益幅・技術一致数・最低スコア等の条件を提供)。



AIで「提案の質」を上げ、辞書で再現性を高める

上位候補をClaude APIで精密分析し、本文から検出した 未知スキルをAIでカテゴリ推定→辞書へ追加入庫。翌日以降の分類・一致判定の精度を「仕組み」で継続改善する。

AI精密分析 (Claude API) 強み/不足/推奨アクションを要約して提案品質を底上げ

分析アウトプット (ウォーターフォール)

AI分析

- 強み抽出
要件と一致する根拠を整理
- 不足/確認点
懸念を事前に潰す質問へ変換
- 次アクション
推奨単価/提案文の材料を提示

提案文補助
文章構成の骨子

推奨単価
レンジを提示

Input : 案件メール+ 要員メール (テキスト) → Output : 根拠/不足/単価/次アクション

未知スキル解決 (辞書拡張) 未登録語を検出→カテゴリ推定→辞書へ追加 (精度の自己増殖)

未知語の処理パイプライン

unkCnt : 0~

1 抽出
本文から候補語を抽出

2 推定
カテゴリ/同義語を推定

3 追加入庫
ユーザー辞書に反映

継続改善
辞書が育つほど、スキル一致・分類精度が安定

移行/バックアップ
辞書エクスポートで運用を引き継ぎ

Note : APIキー未設定時はAI機能を利用不可 (UIで状態表示)

スキル辞書 (14カテゴリ) 表記揺れを正規化し、案件⇔要員のスキル一致判定を安定化

カテゴリ=分類軸 / 正規化=照合精度

- 言語
- FW
- クラウド
- DB
- AI
- NW
- 運用
- テスト
- デザイン
- 手法
- 業界
- 役割
- ツール
- モバイル

表記揺れの正規化
例 : Java8/Java11 → Java
照合ブレを減らしスコアの一貫性を確保

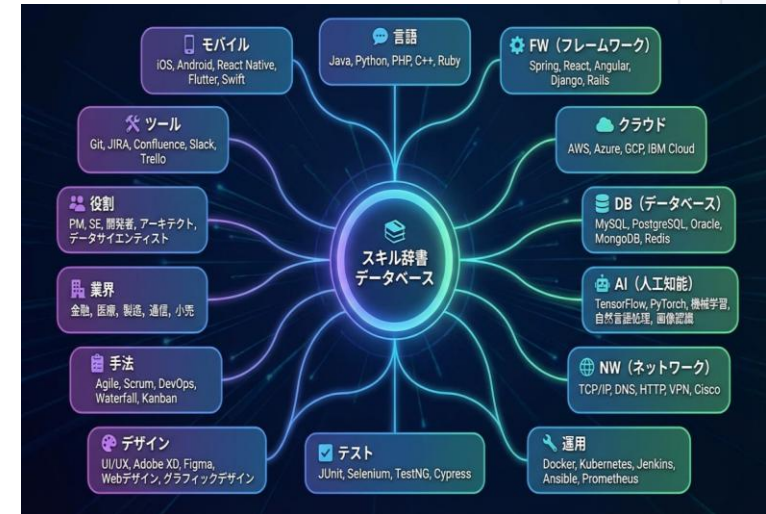
同義語対応
略称・別名を統一して一致判定
例 : サービス名/製品名の揺れを吸収

ユーザー辞書
未知語の追加を蓄積し、次回から自動ヒット
エクスポートでバックアップ可能

運用ポイント
AIは「上位候補の説明責任 (提案根拠)」を補強し、辞書は「一致判定の再現性」を担う。未知語をAIで解決して辞書へ反映することで、マッチング精度を継続的に高められる。

辞書構造 (概念図) カテゴリ別に管理し、スキル一致を安定化

P5



統計
辞書ヒット / 未知語件数を可視化

辞書エクスポート
ユーザー追加語を含めて保存

Outcome : 提案根拠の補強 + 辞書成長でマッチング精度が向上

v6 Pro



結果を「送る・共有する・残す」までワンストップで完結

マッチング結果から、提案メール、Excel/CSV出力、SMTP送信・辞書バックアップまでをUI上のアクションとして提供する。「案件別表示」「絞り込み」「全選択」で提案対象を素早く確定できる。



Excel出力 (.xlsx)

分類/マッチング結果を整理して共有



用途

社内共有・提案管理・履歴保管



実装

SheetJS/xlsx

Sheets : 分類 (案件/要員) ・結果 (詳細)



CSV出力

スコア一覧を軽量にエクスポート



対象

絞り込み後の表示結果



連携

BI/集計ツールに取り込み

Columns : スコア・利益幅・一致数 など



提案メール (生成→送信) / 辞書バックアップ

選択行から下書き生成し、SMTPまたはmailtoで送信。辞書は追加語込みで保全。

One Action



選択

全選択/個別で確定



編集

モーダルで確認・修正



SMTP

STARTTLS/SSL/なし



辞書

ユーザー語込みで出力

Flow : 結果選択 → 生成 → 編集 → 送信/起動

Note : 送受信はブラウザ制約のためHTTPプロキシが前提



UI操作フロー (推奨)

ビュー→絞り込み→選択→出力/送信

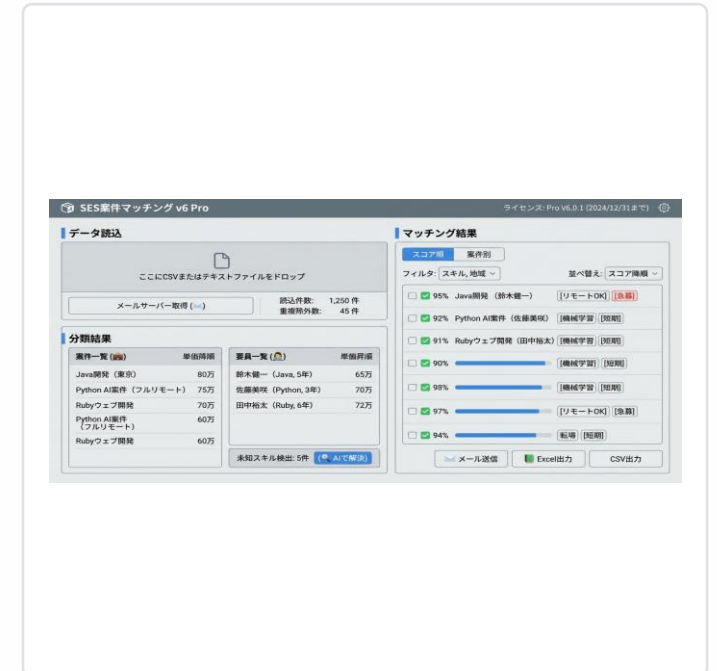
P6

1 ビュー/並べ替え → 2 絞り込み → 3 選択 → 4 出力/送信

v6

UIワイヤフレーム (概念図)

結果リスト → 選択 → 出力/送信 の Wireframe 位置関係を把握



モーダル

詳細表示・メール編集に利用



進捗

処理状況をバー/文言で表示

Outcome : 提案対象を選び、Excel/CSV/メールで即アクション

v6 Pro

注: 送受信はブラウザ制約によりHTTPプロキシが前提。出力は端末上でファイル生成し、共有フローに接続する。