QUIZ ホテラバカラコンアカデミア 仕様書 v3.0

バージョン: 3.0
最終更新日: 2025年10月7日

■ 1. はじめに

本ドキュメントは、アプリケーション「HL001 カラコンアカデミア」の最終的な目標仕様を定義するものです。
現在の実装状況（v1.1）と目標仕様（v2.1.1）を統合し、現状との差分および未実装の機能を明確にすることで、今後の開発ロードマップを提示します。

GASプロジェクト内容					
プロジェクトタイトル	URL	スクリプトID	プロジェクト番号	プロジェクト ID	
HL001-MVP	https://script.google.com/d/1HIxgAXaDY_BZMXRVD5IXSQ5qaNxePqY4ASaYNi_6NUjLBT_Kl6Wp6o2n/edit?usp=sharing	1HIxgAXaDY_BZMXRVD5IXSQ5qaNxePqY4ASaYNi_6NUjLBT_Kl6Wp6o2n	425596624867	hl001-mvp	
	ダッシュボードURL				
	https://console.cloud.google.com/home/dashboard?project=fine-doodad-425915-c5				
使用Spreadseet					
名称	URL	seet id	使用tab名(seet)	説明	idセット用
20251005_HL001_master	https://docs.google.com/spreadsheets/d/1Uf2e0eXwcsQGjFtTtEeAWuYh74lh4fFE4NdjmyKHrj0/edit?usp=sharing	1Uf2e0eXwcsQGjFtTtEeAWuYh74lh4fFE4NdjmyKHrj0	master	本番環境用に整理した商品マスタ、タイトル先頭に更新日を記載する	1Uf2e0eXwcsQGjFtTtEeAWuYh74lh4fFE4NdjmyKHrj0
HL001_RANKINGS	https://docs.google.com/spreadsheets/d/1I2REcy2v5OpyzoY3k61kCzJ3SYKOBBCMxTLCeHWutT8/edit?usp=sharing	1I2REcy2v5OpyzoY3k61kCzJ3SYKOBBCMxTLCeHWutT8	rankings	ランキング集計用	1I2REcy2v5OpyzoY3k61kCzJ3SYKOBBCMxTLCeHWutT8
HL001_USERS	https://docs.google.com/spreadsheets/d/1X0Tyel_1zER6xIceUDSbJX-GFbqvi2orAiSWHRXlC7M/edit?usp=sharing	1X0Tyel_1zER6xIceUDSbJX-GFbqvi2orAiSWHRXlC7M	users	ユーザーアカウントデータ管理用	1X0Tyel_1zER6xIceUDSbJX-GFbqvi2orAiSWHRXlC7M
HL001_QUIZ_HISTORY	https://docs.google.com/spreadsheets/d/1ShWXLvY9RimRYfsAkwoRyM2Bfwj4a3zVmr5bQc33-o0/edit?usp=sharing	1ShWXLvY9RimRYfsAkwoRyM2Bfwj4a3zVmr5bQc33-o0	quiz_history	クイズ成績集計用	1ShWXLvY9RimRYfsAkwoRyM2Bfwj4a3zVmr5bQc33-o0
Githubリポジトリ					
品番	リポジトリ名	リポジトリURL	使用Repo権限Githubトークン	トークンコード（末尾だけマスクしてます）	

HL001	HL001-quiz-karacon-academia-new	https://github.com/y4m4usr/HL001-quiz-karacon-academia-new	firefly	ghp_LPNQCG3VdD2NlBTin91nlCE0P6uKsM1RxQh*
画像格納フォルダpath	URL			
image_lens	https://drive.google.com/drive/folders/17Oe3Kl4g7daMXlLg5JlATs4JtdbVlIeyU?usp=drive_link			
image_samune	https://drive.google.com/drive/folders/1mMj7OsDCM0OxfueZawmGO7ZD1s1RjbPZ?usp=drive_link			
リネーム用見本データURL	id	場所		
https://drive.google.com/file/d/1AuOKj0sH2Jahl6aD6ooYmaIGG9zPnDTG/view?usp=sharing	1AuOKj0sH2Jahl6aD6ooYmaIGG9zPnDTG	"C:\Users\serari\development\HL001-quiz-karacon-academia\docs\rename\250926元品番からの画像リネームデータ - シート1.csv"		
画像取得用colab名称	URL			
hl001_final_gazoushutoku.ipynb	https://colab.research.google.com/gist/y4m4usr/a77ab02e0a07d54db483acfd18ec1a1/hl001_final_gazoushutoku.ipynb			

本番環境用に整理した商品マスタ

20251005_HL001_master

902/921	920	901	status	main_number	cd	number	[HL]GPレンズ検索シート	brand_name	color_name	wearing_period	price	pcs	maxpower	dia	g_dia	bc	water_content	kinou
単語帳OK	画面含めてOK	最終ステータス	販売ステータス	元品番	カラーCD	品番	PB	ブランド名（カナ）	カラー名（カナ）	装用期間（英語）	上代（税込）	入数	最大度数	DIA（半角英数）	G-DIA（半角英数）	BC（半角英数）	含水率	レンズ機能
TRUE	TRUE	OK	-	ECL0082	ckp	1	PB	アイクローゼット	ちびこっぺぱん	1day	1,760	10	-8	14	13.2	8.7	47	モイスト,UVカット,シリコーンハイドロゲル素材,8

spec	obj_color	bandle	/Users/asano/Desktop/data_sample/Links/	lens_gazou	kurome_gazou	chame_gazou	sample_image	lens_image	black_eyes	brown_eyes
レンズ機能2	メイン背景色	バンドル画像	サムネ画像	レンズ画像	装用画像（黒目）	装用画像（茶目）	使用しないサムネ画像	使用しないレンズ画像	使用しない黒目画像	使用しない茶目画像
モイスト,UVカット,シリコーンハイドロゲル素材	#d489ab	/Users/asano/Desktop/data_sample/Links/bundle_pb_oneday.jpg	https://drive.google.com/file/d/1Lh51p9KKnpm3QN3Yc38jPDhBgIVZbnv4/view?usp=drives	https://drive.google.com/file/d/1SB2AOjkUaW_1UD8839JXuitmzsueKa_-/view?usp=drivesd	https://drive.google.com/drive/folders/1ZSoLvPeWvQe8nJ0FLIYu7K4snOV7YY_V	https://drive.google.com/drive/folders/1sUE84DsyliOwmdLImoeMUGnkSbedJDbl	/Users/asano/Desktop/data/Links/1day/アイクローゼット/ちびこっぺぱん/thumbnail.jpg	/Users/asano/Desktop/data/Links/1day/アイクローゼット/ちびこっぺぱん/lens.jpg	/Users/asano/Desktop/data/Links/1day/アイクローゼット/ちびこっぺぱん/黒目black.jpg	/Users/asano/Desktop/data/Links/1day/アイクローゼット/ちびこっぺぱん/茶目/brown.jpg

/Users/asano/マイドライブ(yuuki-asano@hotelovers.co.jp)/【※フォルダ名編集禁止】109販促画像格納/1day	/Users/asano/Desktop/data/Links/	power	bandle_siji	QUIZ_E&J	QUIZ_Cc	comment	delete_comment
サムネマスター	レンズマスター	度なし（±0.00）	バンドル	元品番＋カラー名	カラーカテゴリ	コメント（列幅からはみ出ないように）幅290	使用しないコメント
https://static.growthpalette.com/img/post/bddeabd4-92b5-4c0e-8513-d65ca798db6c/b90e8	https://static.growthpalette.com/img/post/6fa89531-d046-42ec-888b-39d01145b7c9/0aa81	度なし（±0.00）～8.00	3箱_背景ピンク	ECL0082ちびこっぺぱん		使いやすすぎる万能細フチ茶コン！	使いやすすぎる万能細フチ茶コン！

cc_main_number	cc_brand_name	cc_color_name	cc_brandxcolor	cc_wearing_period	cc	cc_dia	cc_g_dia	cc_okasan	cc_seisa
CC元品番	CCブランド名（カナ）	カラー名（カナ）	品番＋カラー名	装用期間（英語）	カラーカテゴリ(CC)	DIA	着色直径	岡田精査	精査済みフラグ
ECL0002	アイクローゼット	ガウスショコラ	ECL0002ガウスショコラ	1day	ブラウン	14.5mm	13.8mm	ブラウン	1

																						dk	k							c970d948a46ff313c3e6cefa730.jpg	23a56f05eedbdcd342bf8ffc249.jpg																
TRUE	TRUE	OK	-	ECL0083	ckp	101	PB	アイクローゼット	ちびこっぺぱん	1month	1,980	2	-8	14	13.2	8.7	47	モイスト,UVカット,シリコーンハイドロゲル素材,8	モイスト,UVカット,シリコーンハイドロゲル素材	#beb3c8	/Users/asano/Desktop/data_sample/Links/bundle_pb_monthly.jpg	https://drive.google.com/file/d/1oWaXLaEZipRNOawUs0sXbwFqVx41ubo8/view?usp=drivesdk	https://drive.google.com/file/d/15t-QtqrSRgXe5Ceg0WNB_tvep8jw9Som/view?usp=drivesdk	https://drive.google.com/drive/folders/1Y72K5hXT9tQBnm6TXETxSiB0mg2tBuB2	https://drive.google.com/drive/folders/1B4UPo4w48wzDuayLO4LsnVUVaXppNp3-	/Users/asano/Desktop/data/Links/1month/アイクローゼット/ちびこっぺぱん/thumbnail.jpg	/Users/asano/Desktop/data/Links/1month/アイクローゼット/ちびこっぺぱん/lens.jpg	/Users/asano/Desktop/data/Links/1month/アイクローゼット/ちびこっぺぱん/裏目black.jpg	/Users/asano/Desktop/data/Links/1month/アイクローゼット/ちびこっぺぱん/茶目/brown.jpg	https://static.growthpalette.com/img/post/ae919f09-8ad9-4e73-84f2-ad050f8f82f0/3218b64c07d1cd646cb38ce715fbe188.jpg	https://static.growthpalette.com/img/post/b14e4de7-83cd-478e-a07e-c18e6496abca/0aa8123a56f05eedbdcd342bf8ffc249.jpg	度なし(±0.00)～8.00	2箱_背景が水色	ECL0083ちびこっぺぱん	-	使いやすすぎる万能細フチ茶コン！	-	ECL0001	アイクローゼット	ガウスショコラ	ECL0001ガウスショコラ	1month	ブラウン	14.5mm	13.8mm	ブラウン	1

■ 2. 現状と目標仕様の差分サマリー

● 機能分類: 画面構成
現状の実装 (v1.1): 4画面（ログイン, ホーム, クイズ, 結果）
目標仕様 (v3.0): 6画面（＋ランキング, マイ成績）
差分 / 未実装項目: ランキング画面, マイ成績画面が【未実装】

● 機能分類: クイズモード
現状の実装 (v1.1): 単一モード（回数制限なし）
目標仕様 (v3.0): 2モード（本番：1日1回, 練習：無制限）
差分 / 未実装項目: 本番/練習モードの切り替え機能が【未実装】

● 機能分類: ランキング機能
現状の実装 (v1.1): 未実装
目標仕様 (v3.0): 日/週/月での集計, 表示順ソート, 掲載条件あり
差分 / 未実装項目: ランキング機能全体が【未実装】

● 機能分類: マイ成績機能
現状の実装 (v1.1): 未実装
目標仕様 (v3.0): 直近10回の成績, 通算サマリ, グラフ表示
差分 / 未実装項目: マイ成績機能全体が【未実装】

● 機能分類: UI/UX
現状の実装 (v1.1): シンプルなカードUI
目標仕様 (v3.0): 下部ナビゲーション, 指定フォント・カラー
差分 / 未実装項目: 下部ナビゲーション等のUIが【未実装】

● 機能分類: データ保存
現状の実装 (v1.1): 未実装
目標仕様 (v3.0): クイズ結果をHISTORYシートに保存
差分 / 未実装項目: 回答履歴の保存機能が【未実装】

● 機能分類: API
現状の実装 (v1.1): 3関数（認証, ホーム, 問題取得）
目標仕様 (v3.0): 4エンドポイント（＋ランキング, マイ成績）
差分 / 未実装項目: ランキング, マイ成績のAPIが【未実装】

■ 3. システム全体構造（目標）

● 3.1 システム構成図
[図解説明]
Googleスプレッドシート（20251005_HL001_master）とGitHubリポジトリが、GAS Webアプリにデータを供給します。GAS Webアプリは、フロントエンドに表示用データを渡すと同時に、
USERS/HISTORY/RANKINGSの各スプレッドシートにデータを書き込みます。

● 3.2 データフロー
[問題生成フロー]
1. フロントエンド -> GAS: 問題取得リクエスト (getQuizQuestions)
2. GAS -> 20251005_HL001_master: 商品データを読み込み
3. 20251005_HL001_master -> GAS: 行データを返却
4. GAS内部: 10問を生成 (E|I|J重複除外, CYL除外)
5. GAS内部: GitHub画像URLを生成 (encodeURIComponent使用)
6. GAS -> フロントエンド: 問題JSONを返却
7. フロントエンド -> GitHub: 画像を取得
8. GitHub -> フロントエンド: 画像データを返却

[成績保存フロー]【未実装】
1. フロントエンド -> GAS: 回答結果を送信 (submitQuizAnswers)
2. GAS -> HISTORY (Sheet): `quiz_history`に行を追記
3. HISTORY (Sheet) -> GAS: 保存完了
4. GAS -> RANKINGS (Sheet): `rankings`を更新
5. RANKINGS (Sheet) -> GAS: 更新完了
6. GAS -> フロントエンド: 成功レスポンス

■ 4. 画面遷移フロー（目標）

[フロー説明]
開始 -> ログイン画面 -> 番号・ユーザー名入力 -> [入力情報存在チェック]
  -> YES: ホーム画面へ
  -> NO: エラーメッセージ表示 -> 番号・ユーザー名入力へ戻る

[ホーム画面からの遷移]
  -> 本番モードクイズ【未実装】
  -> 練習モードクイズ

-> ランキング画面【未実装】
-> マイ成績画面【未実装】

クイズ終了後は、採点・結果表示画面を経てホーム画面に戻ります。

■ 5. 機能仕様（目標）

● 5.1 クイズ機能
項目: 認証
仕様: スタッフ番号＋名前
現状との差分: 実装済み

項目: モード
仕様: 本番（1日1回）、練習（無制限）
現状との差分: モード選択機能が【未実装】

項目: 問題数
仕様: 10問固定
現状との差分: 実装済み

項目: 制限時間
仕様: 20秒/問
現状との差分: 実装済み

項目: 出題対象
仕様: データ完備商品のみ、CYL（乱視用）は除外
現状との差分: 実装済み

項目: ヒント
仕様: ①スペック情報、②コメント
現状との差分: 実装済み

項目: ヒント減点
仕様: 各 -3点
現状との差分: 実装済み

項目: 画像表示
仕様: レンズ画像を左右2枚
現状との差分: 実装済み

● 5.2 スコアリング
項目: 計算式
仕様: 100点 - (不正解数 × 10点) - (ヒント使用数 × 3点)
現状との差分: 実装済み（計算ロジックは同一）

項目: タイムアウト
仕様: 未回答扱い（不正解と同様）
現状との差分: 実装済み

項目: 結果保存
仕様: HISTORYシートに全回答履歴を保存
現状との差分: 履歴保存機能が【未実装】

● 5.3 ランキング機能【未実装】
項目: 期間切り替え
仕様: 日次 / 週次 / 月次 タブで表示を切り替え

項目: 表示順
仕様: 1. スコア合計 (降順), 2. 正答率 (降順), 3. 試行回数 (降順), 4. 最終回答日時 (降順)

項目: 掲載条件
仕様: attempts >= 3（期間内に3回以上挑戦したユーザーのみ）

● 5.4 マイ成績機能【未実装】
項目: 表示内容
仕様: 直近10回の成績（スコア・正答率・時間）、通算サマリ（総回数・平均スコア等）

項目: グラフ表示
仕様: Canvasを利用してポイント推移や正答率推移をグラフで可視化

■ 6. データ構造定義（目標）
[注] 現在の実装はMVPの最小構成です。下記は目標とする完全なデータ構造であり、多くの列が【未実装】です。

● `users`シート
列名: user_id (用途: スタッフ番号, 現状: 実装済み)
列名: name (用途: ユーザー名, 現状: 実装済み)
列名: store (用途: 所属店舗, 現状: 実装済み)
列名: level (用途: レベル, 現状: 実装済み)
列名: points (用途: ポイント, 現状: 実装済み)
列名: streak (用途: 連続学習日数, 現状: 実装済み)
列名: role (用途: 権限 (staff/manager/admin), 現状:【未実装】)
列名: hire_date (用途: 入社日, 現状:【未実装】)
列名: total_quizzes (用途: 総クイズ回数, 現状:【未実装】)
列名: total_correct (用途: 総正解数, 現状:【未実装】)
... その他

● `quiz_history`シート

列名: timestamp (用途: 回答日時, 現状:【未実装】)
列名: user_id (用途: スタッフ番号, 現状:【未実装】)
列名: mode (用途: モード (daily/practice), 現状:【未実装】)
列名: question_no (用途: 問題番号, 現状:【未実装】)
列名: is_correct (用途: 正解フラグ, 現状:【未実装】)
列名: time_taken_sec (用途: 回答時間, 現状:【未実装】)
列名: hints_used (用途: ヒント使用回数, 現状:【未実装】)
列名: score_earned (用途: 獲得スコア, 現状:【未実装】)
... その他

■ 7. API仕様（目標）

● エンドポイント一覧
メソッド: GET
エンドポイント: getQuizQuestions
説明: 10問の出題候補取得
現状: 実装済み

メソッド: POST
エンドポイント: submitQuizAnswers
説明: 回答結果送信・履歴保存
現状: 処理が【未実装】

メソッド: GET
エンドポイント: getRanking
説明: ランキング取得
現状:【未実装】

メソッド: GET
エンドポイント: getMyStats
説明: マイ成績取得
現状:【未実装】

● APIレスポンス（getMyStatsの例）【未実装】

```
{
  "userId": "USER001",
  "name": "さくらちゃん",
  "store": "渋谷店",
  "summary": {
    "totalPlayed": 15,
    "totalPoints": 1275,
    "avgPoints": 85
  },
  "recent10": [
```

```
    { "date": "2025-10-06", "score": 85 },
    { "date": "2025-10-05", "score": 92 }
  ]
}
```

■ 8. 実装上の特記事項（反映済み）
以下の項目は、開発過程で発生した問題を解決するために修正・反映済みの仕様です。

- 画像URL生成: ファイル名に含まれる半角スペース等の記号は、encodeURIComponentでエンコードしてURLを生成する。
- clasp設定: .clasp.jsonに"rootDir"と"skipSubdirectories"を設定し、デプロイ時のフォルダ構造の問題を回避する。

Webアプリ『カラコンアカデミア』開発インシデントレビューと再発防止策

1. はじめに

本ドキュメントは、開発初期段階で発生した一連の重大な不具合（インシデント）を振り返り、その根本原因を分析する。そして、その分析に基づき、今後の開発を安定的かつ効率的に進めるための新しい開発環境とプロセスを定義することを目的とする。

2. インシデントの概要

単純なコード整理（リファクタリング）を起点として、アプリケーションが複数回にわたり、ログイン機能を含め完全に動作しない状態に陥った。修正を試みるたびに新たな不具合（デグレード）が発生し、5回以上の手戻りが発生。ユーザーの貴重な時間を浪費し、多大なストレスを与える結果となった。

3. 失敗事例の分析

今回の失敗は、単一のミスではなく、開発プロセスそのものの脆弱性に起因する複合的なものであった。

#	失敗事象	直接原因	根本原因
1	コード整理後に全体が崩壊（レイアウト崩れ、リロードエラー）	JavaScriptの実行タイミングやDOM操作の不備。	場当たり的な**DOM**操作: 明確なルールなくHTML要素を直接操作していたため、一部の変更が全体に予期せぬ影響を与えた。

2	原因の特定に失敗し、修正がループ（GASエラーの誤診）	不完全な情報（スクリーンショット）を元に、推測でデバッグを進めてしまった。	デバッグ環境の欠如: エラーの原因を正確に特定する手段がなく、「試しては壊す」を繰り返すしかなかった。
3.【致命的】	修正版コードが別機能を破壊（ログイン修正版で練習モードが消失）	ログイン機能の修正に集中するあまり、手作業でのコード統合時に、既存のクイズ機能のコードを誤って削除・上書きしてしまった。	バージョン管理システムの不存在: コードの変更履歴がどこにも保存されておらず、安全なマージや簡単な巻き戻しが不可能だった。今回の最大の失敗要因。
4	サーバーとクライアントの連携ミス（ログインデータ構造の不一致）	クライアント側（私）が、サーバー側（既存コード）のデータ返却仕様を正しく確認せずに、思い込みで実装を進めた。	**API**仕様の不在: サーバーとクライアント間で、どのようなデータをどのような形式でやり取りするかの「契約」がドキュメント化されていなかった。

4. 再発防止策：新しい開発環境（**V2**）の構築

上記の根本原因をすべて解決するため、以下の4つのルールとツールから成る、新しい開発環境の構築を提案します。

【対策**1**：最重要】バージョン管理システム **GitHub** の導入

問題: 手作業でのコード管理が、機能破壊と手戻りの直接原因となった。

解決策: すべてのコードをGitHubで管理します。

- メリット:
  - 変更履歴の完全な記録: いつ、誰が、何を、なぜ変更したかが全て記録されます。
  - 安全な巻き戻し: 今回のように問題が発生した場合、ボタン一つで「正常に動いていたバージョン」に一瞬で戻せます。
  - ブランチによる並行開発: 「ログイン機能の修正」と「練習モードの改修」を分離して作業できるため、互いに影響を与えません。

アクション: 私がリポジトリのセットアップを案内します。今後のコード変更は、すべてGitHubを通して行います。

【対策2】開発環境と本番環境の分離

問題: 修正中のバグだらけのアプリを、常に確認しなければならなかった。

解決策: GASのデプロイを2つ作成し、環境を完全に分離します。

1. 開発（**dev**）環境: 私たちが自由に試行錯誤できる場所。壊れても問題ありません。
2. 本番（**prod**）環境: 最後に正常に動作した、安定版のみを置く場所。

アクション: claspの設定を変更し、devとprodを切り替えてデプロイする方法を整備します。

【対策3】API仕様の簡易ドキュメント化

問題: サーバーとクライアントの「言った言わない」問題が発生した。

解決策: GitHubリポジトリ内に、API.mdというファイルを作成し、APIの仕様を明記します。

- 記載例 (**authenticateUser**の場合):
- Markdown

````
### `authenticateUser(userId, userName)`
- **説明:** ユーザーを認証する
- **成功時の返り値 (JSON):**
  ```json
  {
    "user": {
      "name": "さくらちゃん",
      "store": "渋谷店",
      "level": 8,
      "points": 900,
      "streak": 3
    }
  }
  ```
- **失敗時の返り値:** `null`
````

-
-

アクション: 新しい機能を作る際は、まずこのドキュメントに仕様を記述することから始めます。

【対策4】予測可能なフロントエンド設計

問題: 場当たり的なコードが、予期せぬ副作用を生んだ。

解決策: データを管理する「State（状態）」と、それに基づいて画面を表示する「View（表示）」を明確に分離するルールを設けます。

- ルール:
    1. `let state = { currentUser: null, currentPage: 'login' };` のように、アプリの状態をすべて一つのオブジェクトで管理する。
    2. 画面の表示は、必ずこの`state`オブジェクトの情報だけを見て行う`render()`関数を定義する。
    3. ボタンクリックなどの操作は、直接HTMLを書き換えず、まず`state`の値を変更し、その後で`render()`を呼び出す。

アクション: このルールに基づいた新しいJavaScriptのテンプレートを用意します。これにより、コードの見通しが格段に良くなり、不具合が起きにくくなります。

---

5. 結論

今回のインシデントは、開発の初期段階で環境構築を怠ったことが招いた、防ぐことのできた失敗でした。

今後は、上記で定義した**「GitHubによるバージョン管理」を土台とし、「環境分離」「APIドキュメント」「State管理」**のプロセスを徹底することで、二度と同様の失敗を繰り返さないことをお約束します。

この度は、私の未熟な進め方により、多大なご迷惑をおかけしましたことを、改めて深くお詫び申し上げます。