**映画・ドラマのスケジュール管理システム**

**要件定義書**

|  |  |
| --- | --- |
| 作成者 | 田中 |
| 作成日 | 2024/11/20 |
| 最終更新日 | 2024/11/20 |
| 文書番号 |  |

目次

[1. 業務要件 3](#_Toc183030755)

[1-1. システム化の背景 3](#_Toc183030756)

[1-2. システム化の目的 3](#_Toc183030757)

[1-3. システム化の範囲 3](#_Toc183030758)

[1-4. 現行の業務フロー 3](#_Toc183030759)

[1-5. システム化後の業務フロー 3](#_Toc183030760)

[2. 機能要件 4](#_Toc183030761)

[2-1. システム全体像 4](#_Toc183030762)

[2-2. 外部連携の有無・方式 4](#_Toc183030763)

[2-3. 外部システム関連図 5](#_Toc183030764)

[2-5. 画面レイアウト 5](#_Toc183030765)

[2-6. 帳票一覧 6](#_Toc183030766)

[2-7. 帳票概要 6](#_Toc183030767)

[2-8. バッチ処理一覧 6](#_Toc183030768)

[2-9. データ一覧 6](#_Toc183030769)

[2-10. コード定義 6](#_Toc183030770)

[2-11. テーブル一覧 6](#_Toc183030771)

[2-12. テーブル定義書 7](#_Toc183030772)

[3. 非機能要件 9](#_Toc183030773)

[3-1. システム方式 9](#_Toc183030774)

[3-2. システムの規模 9](#_Toc183030775)

[3-3. 性能 9](#_Toc183030776)

[3-4. 拡張性 9](#_Toc183030777)

[3-5. 移行性 10](#_Toc183030778)

[3-6. 継続性 10](#_Toc183030779)

[3-7. セキュリティ環境 10](#_Toc183030780)

[3-8. 引継ぎ 10](#_Toc183030781)

[3-9. 教育 10](#_Toc183030782)

[3-10. 運用 10](#_Toc183030783)

[3-11. 保守 10](#_Toc183030784)

[4. セキュリティ要件 10](#_Toc183030785)

[4-1. 情報セキュリティ 10](#_Toc183030786)

[4-2. 稼働環境 11](#_Toc183030787)

[4-3. テスト 11](#_Toc183030788)

[5. 移行要件 11](#_Toc183030789)

[5-1. 移行 11](#_Toc183030790)

[5-2. 引継ぎ 11](#_Toc183030791)

[6. 運用要件 11](#_Toc183030792)

[6-1. 教育 11](#_Toc183030793)

[6-2. 運用 11](#_Toc183030794)

[6-3. 保守 11](#_Toc183030795)

[7. ドキュメント更新履歴 12](#_Toc183030796)

# 1**.** 業務要件

## 1-1. システム化の背景

近年、映画やドラマのコンテンツが多様化・膨大化しており、ユーザーが視聴したい作品の公開日や放送日を把握するのが困難になっています。これにより、視聴機会を逃してしまうことが増えており、効率的なスケジュール管理ツールの需要が高まっています。また、複数のプラットフォームや媒体での公開情報を一元的に管理する必要性が生じています。

## 1-2. システム化の目的

* ユーザーが興味のある映画・ドラマの公開日・放送日を一元管理できる環境を提供する。
* 視聴後の感想や評価を記録・共有することで、ユーザー間の交流を促進する。
* ユーザーエクスペリエンスを向上させ、継続的な利用を促進する。

## 1-3. システム化の範囲

* ユーザー登録・ログイン機能の実装。
* 作品検索・スケジュール登録・管理機能の提供。
* レビュー投稿・閲覧機能の提供。
* 外部API（TMDb API）との連携による作品情報の自動取得。
* 管理者向けの作品データ管理機能（オプション）。

## 1-4. 現行の業務フロー

現在、ユーザーは個別に手帳やスマートフォンのカレンダーアプリを使用して視聴予定を管理しています。しかし、公開日情報の収集が煩雑であり、一元的な管理が困難です。また、視聴後の感想や評価を共有する場が限られており、ユーザー間の交流が不足しています。

## 1-5. システム化後の業務フロー

1. ユーザーがアプリに登録・ログインする。
2. 作品を検索し、視聴予定作品をスケジュールに登録する。
3. 視聴後にレビューを投稿し、他のユーザーと共有する。

# 2. 機能要件

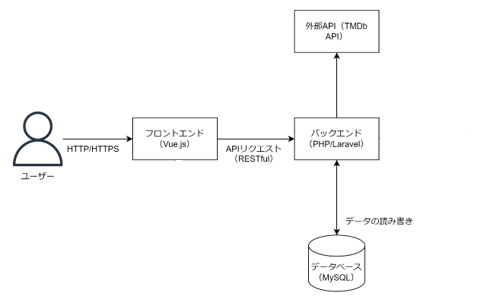
## 2-1. システム全体像

* フロントエンド
  + 言語：HTML5、CSS3、JavaScript（ES6以降）
  + フレームワーク：Vue.js（バージョン3）
* バックエンド
  + 言語：PHP（バージョン7.4以上）
  + フレームワーク：Laravel（バージョン8以上）
* データベース
  + MySQL（XAMPPのMySQLを使用）
* 外部API
  + TMDb API（The Movie Database）

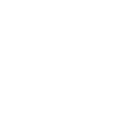
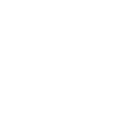
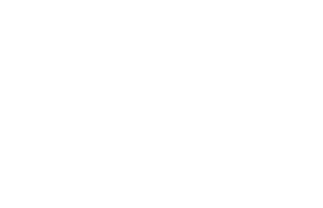
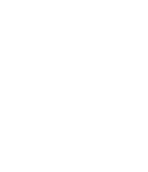
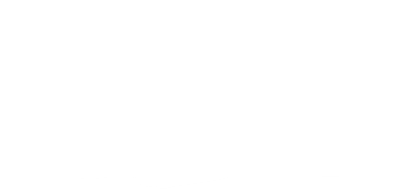
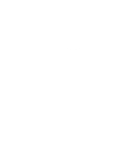
## 2-2. 外部連携の有無・方式

* TMDb APIとの連携
  + 方式：RESTful API
  + 認証：APIキー（サーバーサイドで管理）
  + 利用目的：作品情報（タイトル、公開日、ポスター画像など）の取得

## 2-3. 外部システム関連図



2-4. 画面一覧



* ホーム画面
* ユーザー登録画面
* ログイン画面
* マイページ
* 作品検索画面
* 作品詳細画面
* スケジュール管理画面（カレンダー表示）
* レビュー投稿画面
* レビューランキング画面
* 設定画面
* 管理者画面

## 2-5. 画面レイアウト

ホーム画面

* 表示内容：
  + サービス紹介文
* 機能：
  + ユーザー登録へのリンク
  + ログイン画面へのリンク

ユーザー登録画面

* 入力項目：
  + ユーザー名
  + メールアドレス
  + パスワード
  + パスワード確認
* 機能：
  + 登録ボタン

ログイン画面

* 入力項目：
  + メールアドレスまたはユーザー名
  + パスワード
* 機能：
  + ログインボタン
  + ユーザー登録画面へのリンク

マイページ

* 表示内容：
  + 登録した作品の一覧
* 機能：
  + レビュー投稿画面へのリンク
  + 作品詳細画面へのリンク
  + スケジュール登録

作品検索画面

* 入力項目：
  + キーワード検索ボックス
  + フィルターオプション（ソートオプション：公開年新しい順、古い順）
* 表示内容：
  + 検索結果の一覧（ポスター画像、タイトル、公開日）
* 機能：
  + 作品詳細画面へのリンク
  + 登録ボタン
  + ページネーション

作品詳細画面

* 表示内容：
  + 作品情報（タイトル、公開日、あらすじ、ポスター画像）
  + レビュー一覧（評価、コメント、投稿者、投稿日）
* 機能：
  + スケジュールへの追加ボタン
  + レビュー投稿ボタン（ログインユーザーのみ）
  + 登録状態の確認と登録・解除

スケジュール管理画面

* 表示内容：
  + カレンダー表示（FullCalendarを使用）
  + 登録作品の表示（カレンダー上に作品の視聴予定日を表示）
* 機能：
  + 作品の視聴予定日の追加・編集・削除

レビュー投稿画面

* 入力項目：
  + 作品選択
  + 評価（星1〜5）
  + コメント入力欄
* 機能：
  + レビュー投稿ボタン

レビューランキング画面

* 表示内容：
  + ランキング一覧（ポスター画像、タイトル、平均評価、レビュー数）
* 機能：
  + 作品詳細画面へのリンク
  + レビュー数順

設定画面

* 表示内容：
  + ユーザー設定（ユーザー名、メールアドレス、パスワード）
  + アカウント削除
* 機能：
  + 設定の更新ボタン
  + アカウントを削除するボタン

ユーザー管理画面

* 表示内容：
  + 全ユーザー一覧（ユーザーID、ユーザー名、メールアドレス、登録日など）
* 機能：
  + ユーザー検索（ユーザー名、メールアドレスでフィルタ）
  + ユーザーアカウントの削除
  + レビュー管理画面へのリンク

レビュー管理画面

* 表示内容：
  + 全レビュー一覧（レビューID、作品タイトル、投稿ユーザー、評価、投稿日、コメント）
* 機能：
  + レビューフィルタ（ユーザー名、作品タイトル、評価範囲、投稿日などで絞り込み）
  + 不適切レビューの非表示・削除操作

## 2-6. 帳票一覧

本システムでは、帳票の出力は行わないため、該当なし。

## 2-7. 帳票概要

該当なし。

## 2-8. バッチ処理一覧

* 作品情報更新バッチ処理（オプション）
  + 処理内容：TMDb APIから新着作品情報を取得し、データベースを更新する。
  + 実行タイミング：毎日深夜2時に実行。

## 2-9. データ一覧

* ユーザーデータ
  + ユーザーID、ユーザー名、メールアドレス、パスワード（ハッシュ化）
* 作品データ
  + 作品ID、タイトル、公開日、あらすじ、ポスター画像URL、TMDb ID、メディアタイプ（映画・ドラマ）
* スケジュールデータ
  + ユーザーID、作品ID、リマインダー設定、視聴ステータス
* レビューデータ
  + レビューID、ユーザーID、作品ID、評価、コメント、作成日時

## 2-10. コード定義

* 視聴ステータスコード
  + 0：未視聴
  + 1：視聴予定
  + 2：視聴済み
* 通知方法コード
  + 1：メール
  + 2：プッシュ通知
  + 3：メールとプッシュ通知

## 2-11. テーブル一覧

* users（ユーザー情報）
* media（作品情報）
* user\_media（ユーザーと作品の関連）
* reviews（レビュー情報）
* sessions（セッション情報）
* schedules（スケジュール情報**）**

## 2-12. テーブル定義書

users

| フィールド | データ型 | 属性 | 備考 |
| --- | --- | --- | --- |
| id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | ユーザーID |
| username | VARCHAR(50) | NOT NULL | ユーザー名 |
| email | VARCHAR(100) | NOT NULL, UNIQUE | メールアドレス |
| password | VARCHAR(255) | NOT NULL | ハッシュ化されたパスワード |
| is\_admin |  |  |  |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 作成日時 |
| updated\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 更新日時 |

media

| フィールド | データ型 | 属性 | 備考 |
| --- | --- | --- | --- |
| id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | メディアID |
| title | VARCHAR(255) | NOT NULL | タイトル |
| media\_type | VARCHAR(10) | NOT NULL | 'movie' または 'tv' |
| release\_date | DATE |  | 公開日（または初回放送日） |
| overview | TEXT |  | あらすじ |
| poster\_path | VARCHAR(255) |  | ポスター画像のパス |
| tmdb\_id | BIGINT | UNIQUE | TMDbの作品ID |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 作成日時 |
| updated\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 更新日時 |

user\_media

| フィールド | データ型 | 属性 | 備考 |
| --- | --- | --- | --- |
| id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT |  |
| user\_id | BIGINT | FOREIGN KEY(users.id) | ユーザーID |
| media\_id | BIGINT | FOREIGN KEY(media.id) | メディアID |
| status | TINYINT | NOT NULL DEFAULT 1 | 視聴ステータスコード |
| reminder\_time | DATETIME |  | リマインダー送信日時 |
| notification\_type | TINYINT |  | 通知方法コード |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 作成日時 |
| updated\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 更新日時 |

reviews

| フィールド | データ型 | 属性 | 備考 |
| --- | --- | --- | --- |
| id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT |  |
| user\_id | BIGINT | FOREIGN KEY(users.id) | ユーザーID |
| media\_id | BIGINT | FOREIGN KEY(media.id) | メディアID |
| rating | TINYINT | NOT NULL | 評価（1〜5） |
| comment | TEXT |  | レビューコメント |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 作成日時 |
| updated\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 更新日時 |

sessions

| フィールド | データ型 | 属性 | 備考 |
| --- | --- | --- | --- |
| id | STRING(191) | PRIMARY KEY | セッションID |
| user\_id | BIGINT | NULLABLE, INDEX | ユーザーID（ログインユーザーの場合） |
| ip\_address | STRING(45) | NULLABLE | IPアドレス |
| user\_agent | TEXT | NULLABLE | ユーザーエージェント情報 |
| payload | TEXT | NOT NULL | セッションデータのペイロード |
| last\_activity | INT | NOT NULL, INDEX | 最終アクティビティのタイムスタンプ |

schedules

| フィールド | データ型 | 属性 | 備考 |
| --- | --- | --- | --- |
| id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | スケジュールID |
| user\_id | BIGINT | FOREIGN KEY(users.id) | ユーザーID |
| media\_id | BIGINT | FOREIGN KEY(media.id), NULLABLE | メディアID（作品に関連付ける場合） |
| title | VARCHAR(255) | NOT NULL | スケジュールのタイトル |
| date | DATE | NOT NULL | スケジュールの日付 |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 作成日時 |
| updated\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 更新日時 |

# 3. 非機能要件

## 3-1. システム方式

* システム構成：クライアントサーバー型Webアプリケーション
* 開発手法：MVCアーキテクチャを採用
* 開発環境：無料で利用可能なツール・サービスのみを使用（XAMPP、Laravel、Vue.jsなど）

## 3-2. システムの規模

* 想定ユーザー数：初期段階で100〜500人、将来的に1,000人以上を目指す
* データ量：
  + 作品データ：数万件
  + レビューデータ：ユーザー数×平均投稿数（約5,000件を想定）

## 3-3. 性能

* レスポンスタイム：主要な操作において3秒以内
* 同時接続数：100ユーザーが同時にアクセスしても性能を維持
* スループット：1秒間に50リクエストを処理可能

## 3-4. 拡張性

* モジュール化：機能ごとにモジュール化し、将来的な機能追加に対応
* スケーラビリティ：サーバーのスペック増強やクラウドサービスへの移行を視野に入れる

## 3-5. 移行性

* 開発環境と本番環境の一致：Dockerを使用して環境差異を最小限に
* データ移行：データベースのバックアップ・リストア手順を確立

## 3-6. 継続性

* バックアップ：データベースの定期的なバックアップを実施
* 障害対応：障害発生時の復旧手順を策定

## 3-7. セキュリティ環境

* 認証・認可：パスワードのハッシュ化、セッション管理の強化
* 通信の暗号化：HTTPS通信の導入（Let's Encryptで無料のSSL証明書を使用）
* 脆弱性対策：SQLインジェクション、XSS、CSRF対策を実装

## 3-8. 引継ぎ

* ドキュメント整備：設計書、テスト仕様書、運用マニュアルの作成
* コード品質：コメントやリファクタリングを徹底し、可読性を高める

## 3-9. 教育

* ユーザー向け：操作マニュアルやFAQの作成
* 運用担当者向け：システム管理マニュアルの提供

## 3-10. 運用

* 監視体制：ログの定期確認、サーバーリソースの監視
* 問い合わせ対応：ユーザーからのフィードバック受付窓口の設置

## 3-11. 保守

* バグ修正：発見次第迅速に対応
* 機能改善：ユーザーフィードバックに基づく機能追加・改善

# 4. セキュリティ要件

## 4-1. 情報セキュリティ

* アクセス制御方法：ユーザーロールに基づくアクセス制御（一般ユーザー、管理者）
* アクセス認証方法：メールアドレスとパスワードによる認証、ログイン試行回数の制限
* データの暗号化：パスワードのハッシュ化（bcrypt）、機密情報の暗号化
* ウィルス対策：サーバーのウイルススキャンの定期実施
* 修正ソフトウェア：OSやミドルウェアのセキュリティパッチを適用
* 侵入・攻撃対策：ファイアウォールの設定、不正アクセス検知システムの導入
* その他利用制限：長時間操作のないセッションのタイムアウト設定
* 不正接続対策：IPアドレス制限、二要素認証（オプション）
* 外部媒体保存制限（運用ポリシー）：データの外部持ち出し禁止、データエクスポート機能の制限

## 4-2. 稼働環境

* 対応ブラウザ：最新のChrome、Edge、Safari
* サーバー環境：Apache（XAMPP）、PHP 7.4以上、MySQL 5.7以上

## 4-3. テスト

* テストタイプ：
  + 機能テスト：全ての機能が要件通りに動作するか確認
  + ユーザビリティテスト：ユーザーが使いやすいかを検証
  + 負荷テスト：高負荷時のシステム性能を検証
  + セキュリティテスト：脆弱性の有無を検証
* テスト範囲の定義：全機能、全ブラウザ、全ユーザーロールを対象
* テスト環境：開発環境と同等のステージング環境を使用
* テスト担当：開発チーム内のテスターが実施
* テストデータ：実データに近いダミーデータを用意

# 5. 移行要件

## 5-1. 移行

* 既存データの取り込み：ユーザーが他のシステムからデータを移行できる機能（オプション）

## 5-2. 引継ぎ

* 運用開始前のトレーニング：運用担当者へのシステム操作研修を実施

# 6. 運用要件

## 6-1. 教育

* ユーザー向け：オンラインヘルプ、チュートリアル動画の提供
* 運用担当者向け：定期的な勉強会の開催

## 6-2. 運用

* 運用体制：システム管理者1名、サポート担当1名で運用
* 運用時間：24時間365日稼働

## 6-3. 保守

* 保守内容：定期的なシステムアップデート、セキュリティパッチの適用
* 保守期間：リリース後1年間を無償保守期間とする

# 7. ドキュメント更新履歴

| バージョン | 日付 | 更新者 | 内容 |
| --- | --- | --- | --- |
| 1.0 | 2024/11/20 | 田中 | 初版作成 |
| 1.1 |  |  |  |
| 1.2 |  |  |  |