**カレンダーアプリ開発企画書**

1. **企画書のタイトル**

**・プロジェクト名：カレンダーアプリ**

**・作成日：2024/12/06**

**・作成者：枝尾　雅人**

1. **プロジェクト概要**

**・目的：日常のスケジュール管理を簡単にするアプリを開発し忘れないようにする**

**・背景：どんな方でも扱える個人向けのシンプルなアプリが欲しかったため**

**・目標：ユーザーが直感的に操作し、スケジュールの管理や編が可能なツールを提供する**

1. **ターゲットユーザー**

**・ユーザー層：全ての層**

**・ユーザーニーズ：シンプルなUIとシンプルなスケジュール管理機能**

1. **機能要件：**

**・予定の追加、編集、削除機能**

**・月の切り替え機能**

**非機能要件：**

**・操作性が良いこと**

**・直観的に操作できるUI**

1. **システム構成**

**・アーキテクチャ：フロントエンドとバックエンドを統合したシステム**

**・フロントエンド：HTML、CSS、JavaScript**

**・バックエンド：Python**

**・データベース：Sqlite**

1. **開発スケジュール**

|  |  |  |
| --- | --- | --- |
| **フェーズ** | **期間** | **主なタスク** |
| **設計・要件定義** | **半日** | **機能設計とデータベース設計** |
| **開発** | **2日** | **カレンダー表示、機能の実装** |
| **UIデザイン** | **1日** | **Cssでデザイン調整** |
| **テスト** | **1日** | **単体テスト、統合テスト** |
| **修正・最終調整** | **1日** | **バグ修正と最終確認** |

1. **リソース**

**・使用時間：1日6時間、週12時間程度**

**・進歩管理方法：定期的なテストとタスク調整**

1. **リスクと対策**

**・予測されるリスク：制作時間が短いためクオリティが低いこと**

**・対策：シンプルなデザインにする**

1. **品質管理とテスト**

**・テスト戦略**

1. **単体テスト（各機能の動作確認）**
2. **統合テスト（全体の整合性確認）**
3. **UIテスト（デザインと操作性確認）**

**・品質基準**

**・ユーザーにとって直観的に操作可能なUI**

**・エラーが発生しないこと**

1. **納品物**

**・納品物の概要：完成したカレンダーアプリとマニュアル**

**・納品予定日：2024/12/18**

1. **運用・保守計画**

**・運用サポート：簡易的なメンテナンスとバグ修正対応**

**・保守計画：ユーザーからのフィードバックに基づく機能の追加を検討**

1. **レビューと反省**

**・開発期間が短かったので簡易的な機能しか実装できなかった。**

**・授業で学んだユーザーの登録機能、ログイン機能なども実装したい**

**・タスクが完了したら手動で削除する機能も実装したい**