

お天気アプリ テスト表

テストケース番号	テスト項目	テスト内容	テストデータ	期待結果	実行結果	備考
TC01	都市名入力機能	正しい都市名を入力して天気情報が表示されるか確認する。	都市名: Tokyo	東京の天気情報（気温、天候）が正しく表示される。	合格	正しい都市名でのテスト
TC02	都市名入力機能	存在しない都市名を入力した場合、ボタン下にエラーメッセージが表示されるか確認する。	都市名: InvalidCity	ボタン下に「エラー: city not found」と表示される。	合格	非存在都市でのエラー確認
TC03	都市名入力詳細確認	詳細な地名（市より下位の区）を入力した場合にエラーメッセージが表示されるか確認する。	都市名: 福岡市早良区	ボタン下に「エラー: city not found」と表示される。	合格	詳細すぎる地名の動作確認
TC04	都市名入力詳細確認	市名のみを正しく入力した場合に天気情報が表示されるか確認する。	都市名: 福岡市	福岡市の天気情報（気温、天候）が正しく表示される。	合格	福岡市単位でのテスト
TC05	都市名入力詳細確認	県名（福岡）を入力した場合、ボタン下にエラーメッセージが表示されるか確認する。	都市名: 福岡	ボタン下に「エラー: city not found」と表示される。	合格	都道府県単位でのエラー確認
TC06	入力バリデーション	空欄の都市名を入力した場合、アラートが表示されるか確認する。	都市名: (空欄)	アラートで「都市名を入力してください」と表示される。	合格	空白時の入力バリデーション確認
TC07	天気情報取得機能	天気APIが正しくデータを返しているか確認する。	都市名: London	ロンドンの天気情報（気温、天候）が正しく表示される。	合格	APIからのレスポンス確認

テストケース番号	テスト項目	テスト内容	テストデータ	期待結果	実行結果	備考
TC08	天気情報取得機能	天気APIのレスポンスが遅延・エラー時の挙動を確認する。	都市名: Paris	ボタン下に「エラー: city not found」と表示される。	合格	APIエラーハンドリング確認
TC09	傘の必要性表示機能	天候が雨の場合に「傘が必要」と表示されるか確認する。	都市名: Osaka	天候: 雨 → 「傘が必要です。」と表示される。	合格	天候に依存した判定確認
TC10	傘の必要性表示機能	天候が晴れの場合に「傘は不要」と表示されるか確認する。	都市名: Kyoto	天候: 晴れ → 「傘は不要です。」と表示される。	合格	天候: 晴れ時の動作確認
TC11	UI表示確認	天気情報の各項目（気温、天候）が正しい順序で表示されるか確認する。	都市名: New York	気温、天候が順序通りに表示される。	合格	表示内容と順序の確認
TC12	UI表示確認	天気アイコン未実装であることを確認し、代替テキストが表示されるか確認する。	都市名: Seoul	天候の状態が「雨」や「晴れ」などテキストで表示される。	合格	天気アイコン未実装の確認
TC13	レスポンス対応確認	画面サイズを変更してUIの崩れがないか確認する。（未実装）	デバイス: PC, モバイル	現時点では対応していない。将来対応が必要。	-	未実装機能の将来計画

テスト表作成のポイント

- **網羅性:** アプリの全機能（入力バリデーション、天気情報取得、傘の必要性表示、UI確認など）を網羅し、エラーケースも含めたテスト項目を設定する。
- **再現性:** 各テスト項目には具体的な入力データや手順を記載し、誰でも同じ手順で再現可能な内容にする。
- **実行順序:** 入力バリデーションや基本機能（TC01～TC06）を優先し、その後にAPI動作やUI確認（TC07以降）を実施することで、効率的なテストを進められるようにする。