

VibePlay Test Strategy (2025-11-25)

1. Unit Test Strategy

Scope

- Pure JavaScript logic
- Utility functions
- State management (Zustand)
- Service layer (mapMood, spotify, tmdb)

Tools

- Jest (configured in `jest.config.js`)

Targets

- `src/services/mapMood.js` : Mood mapping logic
- `src/services/spotify.js` : Spotify parameter building, seed capping
- `src/services/tmdb.js` : TMDb API parameter logic
- `src/state/store.js` : Zustand store actions/selectors
- Utility functions in `src/utils/`

Approach

- Isolate functions and modules for direct input/output testing
- Use mocks/stubs for API calls and external dependencies
- Aim for 80%+ coverage on business logic

Example Tests

- Validate `buildParams` output for various mood/intent combinations
- Test `capSeeds` logic for seed limits
- Ensure store actions update state as expected

2. Functional Test Strategy

Scope

- End-to-end user flows
- UI component rendering and interaction
- Navigation and state transitions
- API integration (TMDb, Spotify proxy)
- Platform-specific features (camera, splash screen)

Tools

- React Native Testing Library (component/UI tests)
- Detox (E2E, if feasible)
- Manual testing on simulators/devices

Targets

- Screen components in `src/screens/` (Onboarding, CaptureMood, Intent, Recommendations, Saved)
- UI components in `src/components/` (NeonButton, MoodSelector, RecommendationCard)
- Navigation stack transitions
- API call flows and error handling
- Splash screen and camera permission flows

Approach

- Simulate user interactions and verify UI/state changes
- Test navigation between screens and data flow
- Validate API integration with mock/fake responses
- Manually verify device-specific and native features

Example Tests

- User selects mood, intent, and receives recommendations
- App displays splash screen correctly on both platforms
- Camera permission prompt and fallback logic
- External linking to Spotify

Summary:

Unit tests ensure correctness of core logic and state, while functional tests validate user flows, UI, and integrations for VibePlay.