

Masonicare Native iOS App Proposal (Concrete CMS Backend)

Project overview

Rebuild Masonicare's public website experience as a **native SwiftUI iOS app** that uses the existing

C## Objectives and benefits

- Present Masonicare as a **modern, professional charity** with a polished, accessible mobile presence
- Increase awareness and engagement with core programs, especially interACTION Grants and Disaster-
- Simplify key actions (learn, apply, donate, contact) into a few intuitive mobile flows, backed by content

Phase 1 scope (lean MVP)

Core features

- **Home screen**
 - Highlights: interACTION Grants, Benevolence Fund, Disaster Relief Fund (cards with Learn / Apply /
 - Impact Stories feed sourced from Concrete pages tagged for community stories.[web:26][web:47]
- **Program screens**
 - interACTION Grants: overview, eligibility, steps, and link or native wrapper for the interACTION Grant
 - Benevolence Fund: program description, eligibility, and secure contact options (email, phone, simple -
 - Disaster Relief Fund: list of active appeals and past responses with donate actions.[web:45][web:4]
- **Donate & Contact**
 - Donate: simple selection of fund (Disaster Relief, Benevolence, General) and deep link to existing pay
 - Contact: address, phone, email, map link to the main Masonicare address.[web:4]
- **Lodge support (public content only)**
 - Guides for running interACTION fundraising and accessing banners/branding.[web:26][web:15]
 - Contact info and guidance on working with Regional Masonicare Managers / District representatives if

Technical implementation (lean)

- **Backend**
 - Enable and configure Concrete CMS REST API for read-only public page types and collections.[web:
 - Add only essential custom endpoints (e.g., home summary, program summaries) to keep backend wo
- **App (SwiftUI)**
 - SwiftUI navigation structure (tab bar or single-tab with sections).
 - Networking layer using async/await and JSON decoding for Concrete endpoints.[web:37][web:21]
 - Simple in-memory or lightweight on-disk caching of key content.
 - Optional push notifications for major Disaster Relief appeals if time and budget

allow.[web:45][web:47]## Budget (minimum AUD 20,000 target)

To hit a minimum budget of **AUD 20,000**, the MVP is deliberately lean, uses existing Concrete

content### Effort estimate (tightly scoped)

- Discovery & UX design: ~30 hours (content mapping, flows, basic visual system).^[web:21]^[web:29]
- Concrete CMS API enablement: ~40 hours (enable REST, expose key page types, simple custom end- iOS app development (SwiftUI): ~90–100 hours (core screens, navigation, networking, light caching, ba- QA and App Store submission: ~20 hours (smoke tests, accessibility checks, App Store assets, submis

****Total:**** approximately ****180–190 hours****.

Cost calculation

Assuming a blended rate of ****AUD 110/hour**** (small specialist team or senior individual contractor):

- 180 hours × AUD 110/hour ≈ ****AUD 19,800****.
- 190 hours × AUD 110/hour ≈ ****AUD 20,900****.

This gives a practical budget band of ****AUD 20,000–21,000**** for the Phase 1 app, on the understanding

- Scope remains constrained to public content and simple flows.
- Advanced features (comprehensive offline mode, complex lodge tools, rich

analytics, in-app authentication### Image and reputation impact

- A well-crafted native app positions Masonicare as a forward-looking, well-governed charity that uses m- Showcasing interACTION Grants, Benevolence and Disaster Relief in an accessible, story-led mobile f- Native support for accessibility (Dynamic Type, VoiceOver, high-contrast UI) aligns with serving older a

Next steps

1. Approve Phase 1 scope and the ****AUD 20,000–21,000**** budget band.
2. Run a short technical discovery sprint to confirm Concrete CMS version, API readiness and hosting co3. Finalise UX flows and visual direction, then proceed into implementation, aiming for a 6–8 week delive