UroRads: Frontend UI Specification

Overview

UroRads is a Progressive Web App (PWA) built in React, optimized for mobile use in portrait orientation. The app has three primary screens accessible via a persistent bottom navigation bar. No authentication. No role selection. All users access all screens.

App Structure

Component	Description
Bottom Nav Bar	Persistent footer with 3 buttons: Case | Archive | Add Case
Screen: Case	View and interact with a single case
Screen: Archive	Browse all published cases
Screen: Add Case	Upload and publish new cases (attending workflow)

Bottom Navigation Bar

Persistent across all screens. iOS-style tab bar with icons and labels. Highlights active screen.

Button	Icon Suggestion	Label	Action
1	Image/scan icon	Case	Navigate to Case screen
2	List/folder icon	Archive	Navigate to Archive screen
3	Plus icon	Add	Navigate to Add Case screen

Screen 1: Case

The primary learning screen. Displays one case at a time with integrated chat functionality.

Layout (top to bottom):

- **Header** — Minimal. App title "UroRads" or case number. Optional: Next button here instead of inline.
- **Image Area** — CT slice image displayed prominently. Fixed aspect ratio. Tap to zoom (future feature).
- **Chat Thread Area** — Scrollable. Explanation appears as first bubble (AI message). User questions and AI responses form thread below.
- **Chat Input Bar** — Fixed at bottom (above nav bar). Text input + Send button.
- **Next Button** — Positioned in header OR floating button. Loads next case, clears chat thread.

Behavior:

- On load: Fetch current case (image + pre-generated explanation).
- Explanation displays as first chat bubble (styled as AI message).
- User types question → submits → AI responds with image+explanation in context.
- Chat thread scrolls. Image remains fixed at top (does not scroll away).
- "Next" clears chat, loads new case.

Component Breakdown:

Component	Type	Notes
CaseImage	Image container	Fixed height, contain/cover fit, tap-to-zoom future
ChatThread	Scrollable list	Array of message bubbles (AI or user)
MessageBubble	Styled div	Different style for AI vs user messages
ChatInput	Input + button	Fixed position above nav bar
NextButton	Button	Loads next case, clears thread

Screen 2: Archive

Browse all published cases. List view with minimal metadata.

Layout:

- **Header** — "Archive" title. Optional: Search bar (future).
- **Case List** — Scrollable list of all cases. Each row shows case number + AI-generated short title (3-4 words).

List Item Format:

```
Case #[number] — [AI-generated title]
Example: "Case #14 — Staghorn Calculus Left Kidney"
Example: "Case #27 — Hydronephrosis Grade III"
```

Behavior:

- On load: Fetch list of all cases (ID, title, category).
- Tap row → Navigate to Case screen with selected case loaded.
- List sorted by case number (newest first) or by category (future).

Component Breakdown:

Component	Type	Notes
ArchiveHeader	Header	Title, future: search input

CaseList	Scrollable list	Maps over cases array
CaseListItem	Tappable row	Displays case # + title, navigates on tap

Screen 3: Add Case

Attending workflow for uploading and publishing new cases. Multi-step flow within single screen.

Flow States:

State	UI Elements
1. Capture	Three buttons: "Take Photo" | "Upload File" | "Choose from Library"
2. Prompt	Image preview (thumbnail) + text input for custom prompt + "Generate" button
3. Generating	Loading spinner + "Generating explanation..." text
4. Review	Image (top) + AI explanation (scrollable) + action buttons
5. Published	Success message + "Add Another Case" button

State 4 (Review) — Action Buttons:

- **Edit** — Opens text editor to directly modify explanation text.
- **Regenerate** — Opens prompt input to provide new instruction, then regenerates.
- **Approve** — Saves case permanently, triggers AI categorization, publishes to library.

Component Breakdown:

Component	Type	Notes
CaptureButtons	Button group	Camera, file upload, photo library access
ImagePreview	Thumbnail	Shows selected image
PromptInput	Text area	Optional custom instructions for AI
GenerateButton	Button	Triggers AI explanation generation
LoadingSpinner	Spinner	Shown during AI processing
ExplanationEditor	Text area	Editable AI output (in Review state)
ActionButtons	Button group	Edit | Regenerate | Approve
SuccessMessage	Alert/card	Confirmation + Add Another option

Responsive Behavior

Primary target: Mobile (portrait). App should function on tablet and desktop but mobile is priority.

Breakpoint	Behavior

Mobile (<768px)	Primary layout. Vertical stack. Full-width components.
Tablet (768-1024px)	Same layout, slightly larger touch targets and fonts.
Desktop (>1024px)	Centered container with max-width. Optional: side-by-side image/chat.

PWA Requirements

- Manifest file with app name, icons, theme color.
- Service worker for offline capability (future: cache viewed cases).
- "Add to Home Screen" prompt on supported browsers.
- Viewport meta tag for proper mobile scaling.

Empty States

Screen	Empty State Message
Case (no cases exist)	"No cases yet. Be the first to add one!" + button to Add Case screen
Archive (no cases exist)	"The archive is empty. Add your first case!" + button to Add Case screen

Error States

Error	UI Response
Image upload fails	Toast/alert: "Upload failed. Please try again." + retry option
AI generation fails	Toast/alert: "Generation failed. Please try again." + retry option
Network error (learner)	Toast/alert: "Connection error. Check your network."
Chat message fails	Inline error below input: "Message failed to send. Tap to retry."

Document 3 of N — UroRads Technical Documentation