

Clinic Queue Pass

Capstone 22 · HealthTech · Talodabi Academy

1 · Project overview

PHC wait estimate and virtual queue ticket on phone.

Deliverable	Full mobile MVP in Figma (not code)
Platform	Mobile app · Android-first · portrait
Screens	11 screens + 1 unhappy path (12 frames)
Features	3 core features
Bottom navigation	Clinics · My Queue · Alerts · Profile
Prototype	4 linked flows

Build order: Auth (screens 1–4) → nav tabs (5–8) → core flow (9–11) → unhappy path → link prototypes.

2 · Problem & primary user

Primary health centres across Nigeria have long unpredictable queues. NHS and US clinics use appointments. Nigerian patients travel far and wait hours with sick children.

Primary user

"Grace, a 29-year-old mother in Ilorin who treks to PHC with a feverish toddler."

Geographic scope: National-level

3 · Goals

User goal: Check estimated wait at her clinic and get notified when the queue shortens.

Business goal: Clinics and HMOs pay for queue management and patient flow analytics.

Success: User gets queue number and understands empty data state.

4 · Market context

QUESTION Talodabi Academy	ANSWER
------------------------------	--------

Who pays	Clinics and HMOs pay for queue SaaS; patients use free.
Why not WhatsApp	Patients call reception. Phones busy; no live wait estimate.
Design focus	Virtual queue ticket and wait badge per PHC. Design for low connectivity updates.

Competitors to check:

- **Walk-in and wait** (default)
- **Hospital appointment portals** (tertiary only)
- **WhatsApp clinic groups** (unstructured)

5 · Validate before you design

Interview 3–5 people who match your primary user. Update this brief if interviews contradict it. Do not start hi-fi until you complete the confirmation below.

FIELD	YOUR ANSWER
Validated user (name, age, city)	
Problem in one sentence	
One interview quote	
Success metric for your test	
Primary colour and tone	
One competitor you checked	

Interview prompts

- "Walk me through the last time this happened."
- "What did you use instead? WhatsApp, paper, or nothing?"
- "What would make you trust this on a cheap Android phone?"
- "What would stop you from using this every week?"

6 · Features & scope

v1 · three features

1. See live or staff-updated wait estimate for registered clinics
2. Take a virtual queue number before leaving home
3. Get notified when you are near the front

Out of scope

- Full hospital EMR
- Telemedicine prescriptions
- Ambulance dispatch

7 · Screens

Design every row in wireframe, high-fidelity UI, and prototype (where the flow applies).

#	LAYER	SCREEN	WHAT TO SHOW
1	Auth	Splash Screen	App logo, name, and tagline for Clinic Queue Pass.
2	Auth	Onboarding (1 screen)	Explain the app in one sentence. One Get Started button.
3	Auth	Sign Up Screen	Name, phone number, state, and PIN. No email required.
4	Auth	Sign In Screen	Phone number and PIN. Forgot PIN link.
5	App shell	Clinics (tab)	Search bar and clinic cards with wait-time badges. Show bottom nav with Clinics active.
6	App shell	My Queue (tab)	Active ticket: your number, people ahead, estimated wait. Cancel option. Show bottom nav with My Queue active.
7	App shell	Alerts (tab)	Toggle queue notifications and saved clinic alerts. Show bottom nav with Alerts active.
8	App shell	Profile (tab)	Name, phone, LGA, and saved clinics list. Show bottom nav with Profile active.
9	Core flow	Clinic detail	Wait time, address, Take queue number. Show bottom nav on post-sign-in screens.
10	Core flow	Queue ticket	Your number and people ahead. Show bottom nav on post-sign-in screens.
11	Core flow	Queue · Near front	Notification state: almost your turn. Show bottom nav on post-sign-in screens.

Layers: Auth = 1–4 · App shell = 5–8 · Core flow = 9–11

8 · Prototype flows

Link these in Figma. Examiners should click through without your voice-over.

Flow 1 · App entry

Splash → Onboarding → Sign Up or Sign In → first main tab

Flow 2 • Core task (all three features)

Sign In → Clinics tab → Clinic detail → My Queue tab → Alerts tab

Core flow screens: Clinic detail · Queue ticket · Queue · Near front

Flow 3 • Navigation

Link every nav tab so examiners can tap each bottom nav item. Tabs: Clinics · My Queue · Alerts · Profile

Flow 4 • Unhappy path

Trigger the unhappy path edge case and link to that screen.

9 • Unhappy path

Design this as its own screen state (frame 12):

Clinic has no update today. Show: No live data. Call the clinic or try again later.

Link in **Flow 4**.

10 • Design system

RULE	DETAIL
Colour	You choose. Keep it simple and readable.
Type	You choose. Keep body text easy to read on a phone.
Tone	Design for your user, not for a tech audience.

Reuse on every screen:

- Primary button (active and disabled states)
- Input field (empty, filled, and error states)
- Card or list item used across the app
- Navigation bar with correct labels and icons
- Empty state: what the screen looks like with no data

Deliverables

#	DELIVERABLE	WHAT GOOD LOOKS LIKE
1	Wireframes	Every screen in this brief, plus the unhappy path. Layout and labels clear.
2	High-fidelity UI	Same frames polished. You choose colour and type. Bottom nav on every post-sign-in screen.
3	Prototype	Four linked flows (entry, core task, navigation, unhappy path). Examiners can use the app without your voice-over.
4	AI prototype	Figma Make: at least 2 screens. Share link.
5	Case study (1 page)	Problem · User · Who pays · Competitors · Solution · One design decision
6	Presentation (6–8 slides)	Name · Problem · User · Solution · Key screens · Flow demo · One decision

Before you present

- ☐ Every screen designed in wireframe and hi-fi
- ☐ Unhappy path screen designed and linked in the prototype
- ☐ Four prototype flows linked end to end
- ☐ At least one empty state on a list tab
- ☐ Shared components reused: button, input, card or row, bottom nav
- ☐ Case study names who pays and one competitor you checked
- ☐ You can explain one design decision in critique (why, not what)
- ☐ Completed 5 user interviews or 1 usability test
- ☐ Checked at least one competitor listed under Market context

© Talodabi Academy · For enrolled students · Not for redistribution

