

Fuel Queue Tracker

Capstone 29 · Consumer tech · Talodabi Academy

1 · Project overview

Community fuel queue map. Label reports as user-submitted.

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	Map · Report · 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

Fuel queues during scarcity waste hours. Developed countries rarely queue for petrol; Nigerian drivers circle multiple stations.

Primary user

"Yusuf, a 40-year-old taxi driver in Abuja who loses half a day searching for fuel."

Geographic scope: National-level

3 · Goals

User goal: See community-reported queue length and fuel availability nearby.

Business goal: Station partnerships, ads, and fleet subscriptions.

Success: User reads queue status and submits report.

4 · Market context

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

Who pays	Station ads, fleet subscriptions, and scarcity-season sponsorship.
Why not WhatsApp	Drivers drive station to station guessing. Radio news is late.
Design focus	Community queue map with last updated time. Design trust into reports.

Competitors to check:

- **Drive and ask** (default)
- **Twitter/X queue posts** (unstructured)
- **Station staff word of mouth**

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. Map of nearby stations with queue status
2. Report queue: Short, Medium, Long, No fuel
3. Alert when a favourite station has short queue

Out of scope

- Fuel delivery ecommerce
- Official NNPC API without partner
- Payment in app

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 Fuel Queue Tracker.
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	Map (tab)	Map with station pins colour-coded by queue length. Show bottom nav with Map active.
6	App shell	Report (tab)	Pick station and queue status. Submit and thank you note. Show bottom nav with Report active.
7	App shell	Alerts (tab)	Favourite stations and notify-when-short toggles. Show bottom nav with Alerts active.
8	App shell	Profile (tab)	Name, city, vehicle type, and report count. Show bottom nav with Profile active.
9	Core flow	Station detail	Queue level, last report time, Report button. Show bottom nav on post-sign-in screens.
10	Core flow	Report · Thank you	Queue status update submitted. Show bottom nav on post-sign-in screens.
11	Core flow	Alert · Short queue	Favourite station now short queue notification. 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 → Map tab → Station detail → Report → Alerts tab

Core flow screens: Station detail · Report · Thank you · Alert · Short queue

Flow 3 • Navigation

Link every nav tab so examiners can tap each bottom nav item. Tabs: Map · Report · 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):

No reports in 24 hours. Show: No recent reports. Be the first to update this station.

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

