

NEPA Light Alert

Capstone 11 · Consumer tech · Talodabi Academy

1 · Project overview

You are designing **community power reports** for one area. Copy must say neighbors reported, not official DISCO schedule. History and alerts are the product.

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	Bottom navigation with 4 items: Status · Report · Alerts · History
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

Grid power in many Nigerian cities often lasts only 4 to 8 hours per day (varies by area). People run generators at full cost waiting for grid power that may not come. There is no official app or system to know when power will be restored to your area. People find out only when the lights finally come on.

Primary user

"Kunle, a 30-year-old freelance designer in Ikeja who works from home and spends over ₦15,000 a month on generator fuel because he cannot predict when NEPA power will be available."

Geographic scope: National-level

3 · Goals

User goal: Know in advance when power is reported in his area so he can plan his work hours and reduce generator costs.

Business goal: Build a crowdsourced power intelligence platform and sell aggregated data to energy companies and businesses.

Success: In a quick test, someone **reports power**, reads **area status**, and sets **one alert** confidently.

4 · Market context

QUESTION	ANSWER
Who pays	Kunle might pay little. Estates, generator services, or energy data buyers are B2B payers (assumption).
Why not WhatsApp	Estate WhatsApp groups already post "NEPA is on." Pain is history, alerts when away, and quiet hours , not one more message in a noisy group.
Design focus	Copy must say " Neighbors reported ", not "official NEPA time." Design area history chart and Report Now as hero actions.

Competitors to check:

- **NEPAWatch, LightDey** (crowdsourced outage maps and alerts)
- **NERC Power Outage Report** (official reporting app on Play Store)
- **Estate WhatsApp / Twitter** (informal, fragmented)
- **eFactory / estate prepaid meters** (billing, not grid-up alerts)

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 community-reported power status for your area right now
2. View your area's average daily power hours over the past week
3. Set an alert to be notified when power comes on in your area

Out of scope

- Generator sales or payments
- Government API claims without a partner
- Billing or meter recharge

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 name and logo. Use a bold electricity metaphor: bright bulb or lightning bolt.
2	Auth	Onboarding (1 screen)	Explain how community reporting works: 'When your neighbours report power, you get notified.' One 'Join the Community' button.
3	Auth	Sign Up Screen	Name, phone number, state, LGA, and area/estate name. PIN for login.
4	Auth	Sign In Screen	Phone number and PIN.
5	App shell	Status (tab)	Large status indicator: Power ON (green) or Power OFF (red). Last reported time. A 'Report Now' button prominently placed.
6	App shell	Report (tab)	A bar chart showing average power hours per day for the past 7 days in the user's area. Simple and clear.
7	App shell	Alerts (tab)	Two large buttons: 'Power is ON' and 'Power is OFF'. After reporting, show 'Thank you. Your neighbours have been notified.'
8	App shell	History (tab)	Toggle to enable power-on alerts. Quiet hours selector (e.g. don't notify between 11pm and 6am).

9	Core flow	Report submitted	Power ON/OFF report thank you state. Show bottom nav on post-sign-in screens.
10	Core flow	Area history	7-day power hours chart for area. Show bottom nav on post-sign-in screens.
11	Core flow	Alert enabled	Power-on alert toggle saved. 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 → Status tab → Report → Report submitted → Alerts tab

Core flow screens: Report submitted · Area history · Alert enabled

Flow 3 · Navigation

Link every nav tab so examiners can tap each bottom nav item. Tabs: Bottom navigation with 4 items: Status · Report · Alerts · History

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 from the user's area in the past 48 hours. Show: 'No recent reports for your area. Be the first to report and help your neighbours.' Show the Report button prominently.

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

11 • Submit & defense

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

