

Cold Chain Alert

Capstone 20 · AgriTech · Talodabi Academy

1 · Project overview

You are designing a **temperature monitor** for one trip. Sensor + disconnect alert are the hero screens. This is B2B hardware story; say that in case study.

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: Monitor · Alerts · Trips · Settings
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

Nigeria loses a large share of food to post-harvest spoilage each year (figures vary by crop; verify any % in your case study). Tomatoes, fish, pepper, and poultry spoil in transit because there is no affordable temperature monitoring system for small distributors. Losses are silently absorbed as the cost of doing business. If you cite spoilage percentages, name **one source** you can defend (crop and year matter).

Primary user

"Alhaji Danjuma, a 55-year-old produce distributor in Kano who trucks tomatoes from farms to markets across three states and loses ₦200,000+ per month to spoilage during transit."

Geographic scope: National-level

3 · Goals

User goal: Monitor the temperature inside his truck in real time and get immediately alerted when it rises above safe levels.

Business goal: Sell the app alongside low-cost IoT temperature sensors to agro-distributors and cold storage operators.

Success: In a quick test, someone reads **live temperature**, understands **danger alert**, and interprets **sensor lost** yellow state.

4 • Market context

QUESTION	ANSWER
Who pays	Alhaji pays for sensor hardware + app bundle , not app alone. Agro logistics, cold storage operators B2B (assumption).
Why not WhatsApp	Drivers notice spoilage at delivery. WhatsApp cannot show live temperature line or sensor disconnect (in brief).
Design focus	Design sensor pairing, trip log, and yellow disconnect alert . App is a monitor; hardware is part of the story.

Competitors to check:

- **Figorr** (formerly Gricd; IoT temp monitoring + alerts; Nigeria)
- **Tranzit cold chain** (fleet telematics; pharma/food grade)
- **Cold Sylos** (refrigerated trucks + monitoring; B2B logistics)
- **Open truck, no sensor** (default for small distributors)

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. Real-time temperature display from a connected IoT sensor
2. Immediate alert when temperature exceeds the safe threshold for the selected produce type
3. Trip log showing the full temperature history for each delivery

Out of scope

- Selling produce or routing trucks
- Fleet management for 100 trucks
- Sensor hardware design (only pairing + readout UI)

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	Clean and functional. Cold blue-green palette to suggest freshness and temperature.
2	Auth	Onboarding (1 screen)	Explain how the app works with a sensor: 'Connect your sensor → Monitor your cargo → Get alerts when temperature rises.' One 'Set Up My Sensor' button.
3	Auth	Sign Up Screen	Business name, owner name, phone number, state, and PIN.
4	Auth	Sign In Screen	Phone number and PIN.
5	App shell	Monitor (tab)	Large temperature reading in the centre of the screen. Current status: Safe (green) or Warning (red). Produce type and trip name shown at the top. Time elapsed.
6	App shell	Alerts (tab)	A line graph showing temperature changes over the current trip duration. A horizontal red line marks the danger threshold. Simple and readable.
7	App shell	Trips (tab)	Full-screen red alert. Temperature reading in large text. Message: 'Temperature too high. Take action now.' Checklist of immediate steps: Check ventilation, Reduce load, Contact destination.

8	App shell	Settings (tab)	List of past trips with produce type, route, date, and a pass/fail indicator (Did it stay in safe range?). Tap a trip to see its full temperature graph.
9	Core flow	Trip · Monitoring	Live temperature reading and safe/warning status. Show bottom nav on post-sign-in screens.
10	Core flow	Temperature graph	Line graph with danger threshold marked. Show bottom nav on post-sign-in screens.
11	Core flow	Alert · Take action	Full-screen warning with action checklist. 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 → Monitor tab → Monitoring → Alert screen → Trips tab

Core flow screens: Trip · Monitoring · Temperature graph · Alert · Take action

Flow 3 · Navigation

Link every nav tab so examiners can tap each bottom nav item. Tabs: Bottom navigation with 4 items: Monitor · Alerts · Trips · Settings

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):

The sensor disconnects mid-trip. Immediately show a yellow alert: 'Sensor signal lost. Last recorded temperature was [X]°C at [time]. Check your sensor connection.' Do not assume the temperature is safe.

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

