

Salary Stretch

Capstone 17 · FinTech · Talodabi Academy

1 · Project overview

You are designing **envelope budgeting** for a fixed monthly salary in naira. Category bars and overspend clarity beat charts and investment features.

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: Budget · Log · Insights · 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

Many Nigerian salary earners run out of money within two weeks of payday. They spend heavily at month start with no plan, then borrow or go without by the 20th. Basic budgeting is not taught in schools and most budget apps feel complex, technical, or designed for a different economy.

Primary user

"Tunde, a 27-year-old civil servant in Abuja earning ₦85,000 a month who has nothing left by the 20th and regularly borrows from colleagues to cover transport and food."

Geographic scope: National-level

3 · Goals

User goal: Set a simple monthly budget and see clear warnings before he overspends on any category.

Business goal: Build a daily-use budgeting habit among Nigerian salary earners and monetise through premium insights and financial product referrals.

Success: In a quick test, someone **sets limits, logs one expense**, and understands **80% warning or overspend** state.

4 • Market context

QUESTION	ANSWER
Who pays	Tunde expects free budgeting. Money is in loan referrals, premium insights, or bank partnerships (assumption).
Why not WhatsApp	People track spend mentally or in notes. WhatsApp does not show category limits at 80% or salary-day planning.
Design focus	Salary in, category bars, overspend rebalance (edge case). Naira-first, civil servant tone.

Competitors to check:

- **MoniMata** (zero-based budgeting; bank alert email capture)
 - **Kuda, OPay, bank apps** (transactions; weak envelope UX)
 - **PiggyVest, Cowrywise** (savings/invest; not daily spend envelopes)
 - **Notebook** (default)
-

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. Enter monthly salary and set a spending limit for each category (food, transport, rent, airtime, savings)
2. Log daily expenses against each category
3. Alert notification when a category reaches 80% of its budget limit

Out of scope

- Investments, crypto, or credit scores
- Business expense tracking
- Automatic bank sync (design manual log only)

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 a simple visual of a coin or wallet. Clean and trustworthy.
2	Auth	Onboarding (2 screens)	Screen 1: 'Your salary should last 30 days. Let's make sure it does.' Screen 2: Enter your monthly salary to get started.
3	Auth	Sign Up Screen	Name, phone number, and PIN.
4	Auth	Sign In Screen	Phone number and PIN.
5	App shell	Budget (tab)	Show total salary, total spent so far, and remaining balance. Below, show each category as a progress bar: category name, amount spent, and limit.
6	App shell	Log (tab)	Form to set spending limits per category. Show the salary at the top and a running total of limits set so far so the user can see what's left to allocate.
7	App shell	Insights (tab)	Amount input, category picker (dropdown or chips), optional note, and date (auto-filled to today). One 'Save' button.
8	App shell	Profile (tab)	Show all transactions in a selected category. Total spent, limit, and remaining. Option to edit the category limit.
9	Core flow	Set budget · Saved	Category limits saved against salary. Show bottom nav on post-sign-in screens.

10	Core flow	Log expense · Saved	Amount logged to one category. Show bottom nav on post-sign-in screens.
11	Core flow	Category · Near limit	Progress bar at 80%+ with warning. 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 → Budget tab → Set budget → Log tab → Insights tab warning

Core flow screens: Set budget · Saved · Log expense · Saved · Category · Near limit

Flow 3 · Navigation

Link every nav tab so examiners can tap each bottom nav item. Tabs: Bottom navigation with 4 items: Budget · Log · Insights · 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):

User has overspent a category. Do not hide it. Show the overage clearly in red. Display: 'You've gone ₦2,400 over your Food budget. You can borrow from your Airtime budget if needed.' Show a simple rebalance option.

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

