

Hostel Room Finder

Capstone 37 · PropTech · Talodabi Academy

1 · Project overview

Student housing listings with verification badge.

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	Search · Saved · 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

University towns have agent scams and housing panic. US has dorm systems; Nigerian students hunt via WhatsApp groups.

Primary user

"Bola, a 20-year-old student at UNILAG searching for affordable hostel space."

Geographic scope: National-level

3 · Goals

User goal: Browse verified listings with price, distance, and landlord contact.

Business goal: Listing fees from landlords; premium verified badge subscriptions.

Success: Student filters campus and saves listing.

4 · Market context

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

Who pays	Landlords pay listing fees; verified badge subscription.
Why not WhatsApp	Students search WhatsApp groups with agent scams.
Design focus	Campus filter, budget, verified badge, save favourites.

Competitors to check:

- **WhatsApp groups** (default)
- **Campus bulletin** (offline)
- **Generic property apps** (not student-focused)

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. Search hostels by school and budget
2. Listing detail with photos and rules
3. Save favourites and contact landlord in-app

Out of scope

- Rent payment escrow without license
- Every Nigerian campus v1
- Landlord background checks you cannot show

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 Hostel Room Finder.
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	Search (tab)	School picker, budget slider, and result cards. Show bottom nav with Search active.
6	App shell	Saved (tab)	Favourite listings with price and distance. Show bottom nav with Saved active.
7	App shell	Alerts (tab)	New listing alerts for saved search criteria. Show bottom nav with Alerts active.
8	App shell	Profile (tab)	Student name, school, and contact phone. Show bottom nav with Profile active.
9	Core flow	Listing detail	Photos, price, rules, Contact button. Show bottom nav on post-sign-in screens.
10	Core flow	Saved · Listing	Listing added to favourites. Show bottom nav on post-sign-in screens.
11	Core flow	Alert · Set	Budget alert created for new listings. 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 → Search tab → Listing detail → Saved tab → Alerts tab

Core flow screens: Listing detail · Saved · Listing · Alert · Set

Flow 3 · Navigation

Link every nav tab so examiners can tap each bottom nav item. Tabs: Search · Saved · 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 listings in budget. Show: No rooms in this range. Widen budget or set alert.

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

