FIXLY

Product Design Brief

Home Maintenance Platform

ASSIGNED DESIGNER	**Ali**
PRODUCT	Fixly
PLATFORM	Mobile Application
PROJECT TYPE	Product Design Internship Project

PROJECT FILE

Use the same Figma account you used during the bootcamp. Create a new file named Fixly - Product Design - Ali and send the share link to your instructor as soon as the file is created.

01 / PRODUCT CONTEXT

Fixly

Fixly is a mobile home maintenance platform that connects residents with service professionals and manages each repair from the first request to completion.

The product consolidates service discovery, bookings, property information, repair requests, quotes, visit updates, payments and service history into one structured experience. The objective is not to reproduce a directory of providers; it is to redesign the experience around the decisions and tasks residents perform when maintaining their home.

Primary user

Homeowners and renters managing repairs, routine maintenance and service visits across one or more properties.

Core problem

Residents currently move between calls, messaging, notes and receipts to find a provider, agree on costs and track a repair.

Product role

Fixly acts as the resident's maintenance hub: what needs attention, who is coming, what changed and which action is required.

Primary platform

The main product is a mobile application. Desktop and the provider-facing platform are outside the required delivery.

PRODUCT PRINCIPLE

> A resident should be able to open Fixly and understand the next service visit and any action required in under a minute.

SUCCESS TEST The interface should reduce repeated calls, unclear costs and missed visits. Every primary screen must make the next relevant action obvious.

Required product areas

The following modules define the required scope. You may refine the structure, but removing or merging a module must be supported by a clear UX decision.

AREA	REQUIRED FUNCTION
Home	Upcoming visits, active repairs, pending quotes, provider updates and time-sensitive actions.
Bookings	Daily and weekly visit schedule, property addresses, arrival windows and cancelled or rescheduled visits.
Services	Service categories and service detail with provider information, coverage, availability and pricing basis.
Repair Requests	All maintenance requests organized by urgency and status, including issue details, photos and progress.
Quotes & Payments	Itemized estimates, approval decisions, confirmed charges, payment status and receipts.
Provider Updates	Visit-specific communication with clear unread and priority states and links to affected bookings.
Maintenance Plans	Discover, view, save and activate recurring maintenance packages for a selected property.
Notifications	Central notification stream for visit changes, quotes, repair progress, payments and plan reminders.
Profile & Settings	Resident information, saved properties, contact details, preferences and notification controls.

NAVIGATION You are responsible for defining the final information architecture and primary navigation. Do not assume that every module needs a permanent navigation item.

What each area must support

Home

- Show the next service visit with arrival window and property address.
- Surface pending quotes and repair requests that need attention.
- Expose important unread provider updates.
- Show recent completion or payment activity without overwhelming the dashboard.
- Provide useful shortcuts based on context.

Bookings

- Support day and week views.
- Show service, provider, arrival window and property context.
- Represent cancelled and rescheduled visits clearly.
- Allow the user to open the related request or visit details.

Repair Requests

- Separate awaiting review, scheduled, in progress and completed requests.
- Show issue description, property, urgency, photos and related quote.
- Make status and the next responsible party immediately understandable.
- Handle a revised quote after the resident has already viewed the original.

Services

- Show available service categories and relevant providers.
- Provide service-level access to scope, availability, provider details and booking.
- Keep repeated service and provider information consistent across the product.

Provider Updates

- Distinguish unread, read and high-priority messages.
- Identify the provider and the booking or request being discussed.
- Support deep-linking to the affected visit, request or quote when applicable.

Maintenance Plans

- Browse recurring packages with scope, frequency, price and renewal terms.
- Open a plan detail page and choose the relevant property.
- Activate or save a plan with explicit confirmation of recurring charges.
- Reflect the first confirmed visit in Bookings and the active plan in the profile.

Flows that must be designed end-to-end

These scenarios are mandatory. The final prototype should demonstrate the complete path, not only isolated destination screens.

#	SCENARIO	EXPECTED FLOW
01	Prepare for a visit	Resident opens Fixly, understands today's bookings, identifies the next provider visit and opens its service and request details.
02	Approve a repair quote	Resident notices a pending quote, opens the itemized estimate, reviews scope and total cost, approves it and sees the request status update.
03	Respond to a change	Resident receives a provider update that a visit is cancelled or moved, reviews the change and verifies the updated booking or selects a replacement slot.
04	Check a completed repair	Resident receives a completion notification, reviews the work summary and confirmed charge, completes payment and opens the receipt in service history.
05	Activate a maintenance plan	Resident discovers a maintenance package, reviews inclusions and recurring terms, selects a property and available first visit, confirms activation and sees it in Bookings.

PROTOTYPE STANDARD

The prototype must make these flows understandable without a verbal walkthrough. Include the decision points, feedback, confirmation and state changes that make each scenario believable.

Design the product outside the perfect scenario

Fixly must remain understandable when information changes, is missing, fails to load or requires immediate attention.

STATE	MINIMUM EXPECTATION
Empty	No upcoming visits, no repair requests, no provider updates, no active maintenance plans.
Loading	Key dashboard modules, lists and detail pages should not jump unpredictably while content is loading.
Offline / connection issue	Preserve context, identify outdated information and prevent unconfirmed actions from appearing successful.
Error	Provide a recoverable action when a request, plan activation or payment cannot be completed.
Quote awaiting approval	Distinguish requests blocked by a resident decision from requests awaiting provider action.
Completed / paid	Show completed work and successful payment without making the request appear unfinished.
Visit cancelled	Keep the schedule readable, make cancellation obvious and offer a clear rebooking path.
Visit rescheduled	Show the new time or arrival window and preserve enough context to explain the change.
Unread / priority	Differentiate routine messages from visit changes or issues requiring immediate resident attention.
Plan activation success	Confirm the plan, recurring terms and first visit; reflect the result elsewhere when relevant.

Notification priority

Do not give every notification equal visual weight. A provider arriving in ten minutes, a revised quote awaiting approval and a routine maintenance reminder have different urgency and should be treated accordingly.

Required work before high-fidelity UI

The project file must show the decisions that led to the final interface. Decorative research pages without a connection to the product will not be considered useful evidence.

Problem framing

Define the current fragmented home maintenance experience and identify the highest-frequency tasks Fixly must improve.

User conversations

Speak with homeowners and renters and capture recurring behaviors, frustrations, workarounds and information priorities.

Benchmark review

Review at least three relevant service-booking or home maintenance products. Document patterns worth keeping and problems worth avoiding.

User profile

Create one concise primary user profile grounded in your findings. Include only details that influence product decisions.

Information architecture

Define the hierarchy, navigation and relationships between modules before visual design.

Flow mapping

Map the five mandatory scenarios and any additional flow required by your chosen structure.

Wireframes

Resolve hierarchy, navigation and content priority at low fidelity before moving to final UI.

REVIEW GATE Do not begin final visual design until the information architecture, mandatory flows and core wireframes are present in the file and ready for review.

Build a product system, not a set of screens

The final interface should behave as one coherent product. Repeated patterns must be designed once and reused through components and variants.

Typography

Define a type scale for display, page title, section title, body, label, metadata and supporting text.

Color

Create semantic color roles for brand, neutral surfaces, success, warning, error and information states.

Spacing & layout

Use a consistent spacing system, grid logic and predictable section rhythm across the entire application.

Components

Build reusable navigation, buttons, fields, tabs, cards, list rows, chips, alerts, modals and status patterns.

Variants

Use variants for meaningful states such as selected, unselected, disabled, unread, awaiting approval, paid, cancelled and priority.

Auto Layout

Core screens and components must be built with Auto Layout so content changes do not break the interface.

Accessibility

Maintain readable contrast, usable touch targets, clear state communication and text that remains understandable without color alone.

Prototype behavior

Use transitions only where they clarify navigation, hierarchy or feedback. Interaction should never hide the current state.

Required interface coverage

- Home dashboard
- Bookings / visit schedule
- Service category list
- Service and provider detail
- Repair requests list
- Repair request detail
- Quotes, payments and receipts
- Provider updates
- Maintenance plans
- Plan detail and activation
- Notifications
- Profile and settings
- Required confirmation, error and empty states

How the project will be reviewed

Keep all work in the same Figma file so progress and decisions remain visible throughout the project.

CHECKPOINT	WHAT MUST BE READY
01 - Discovery	Problem framing, resident insights, benchmark review and primary user profile.
02 - Structure	Information architecture, navigation proposal and mandatory user flows.
03 - Wireframes	Low-fidelity coverage of the main product and all five mandatory scenarios.
04 - Direction	Visual direction, type, color, layout logic and first reusable components.
05 - Product UI	Complete high-fidelity screens, component states and responsive content behavior where relevant.
06 - Prototype	Clickable mandatory flows with realistic content, feedback and state transitions.
07 - Final handoff	Organized Figma file, presentation section and final share link.

Final file should contain

- Brief / project overview
- Discovery and benchmark findings
- Information architecture and user flows
- Wireframes
- Design system and reusable components
- High-fidelity product screens
- Application states
- Clickable prototype
- Final presentation / case summary

09 / FILE SETUP & SUBMISSION

Start here, Ali

Use the same Figma account you used during the bootcamp. The complete project must remain in one file owned by that account.

STEP	ACTION	REQUIREMENT
1	Create the file	Create a new Figma Design file in your existing bootcamp account.
2	Name it exactly	Fixly - Product Design - Ali
3	Create page structure	Recommended pages: 00 Brief, 01 Discovery, 02 UX, 03 Wireframes, 04 Design System, 05 UI, 06 Prototype, 07 Presentation.
4	Set sharing	Enable link access so the instructor can open the file and review progress.
5	Send the link	Send the Figma file link to your instructor immediately after setup. Continue all project work inside the same file.

Acceptance criteria

Product clarity

A first-time user can understand the main navigation and what requires attention without explanation.

UX reasoning

Major decisions are traceable to research, product priority or a clear usability rationale.

Flow completeness

The five mandatory scenarios work end-to-end and include feedback, confirmation and state changes.

UI consistency

Typography, spacing, components, states and content behavior are consistent across modules.

Figma quality

The file uses Auto Layout, components, variants and organized naming rather than disconnected frames.

Handoff quality

The final file is clean enough that another designer or developer can understand the product structure without rebuilding it.

FIRST ACTION Create the Figma file using your existing bootcamp account and send the link to your instructor before beginning the design work.