

UNIT 5: UI DESIGN CONSIDERATIONS · §5.4

Navigational Models in UI/UX

Architectural topologies for wayfinding and information space:

Hub & Spoke · Fully Connected · Multilevel Tree · Stepwise · Pyramid · Flat

BCA 2nd Semester

Information Architecture

Real Website Case Studies

Pros & Cons Analysis

What is a Navigational Model?

A **Navigational Model** defines how screens and content are structured, determining how users move through an application based on task complexity and information architecture.

Why Architectural Topology Matters

Navigation models reflect the user's **mental model**. Choosing the wrong model forces unnatural cognitive overhead:

- **Wayfinding Confidence:** Prevents users from getting lost or "pogo-sticking".
- **Backtracking Overhead:** Dictates whether users need 1 click or 4 clicks to switch tasks.
- **Task Efficiency:** Channels user focus in high-stakes workflows (e.g. checkouts) vs open-ended exploration (e.g. wikis).

The 3 Core Architectural Dimensions

1. Centralization: Central origin (Hub) vs Decentralized (Mesh) vs Distributed peers (Flat).

2. Depth vs Breadth: Deep ancestral tiers (Tree/Pyramid) vs Single-tier direct access (Flat).

3. Traversal Constraints: Freeform associative jumping vs Strict linear funnel (Stepwise).

The 6 Primary Navigational Models

Each model serves distinct cognitive patterns and operational constraints:

Model 1

Hub & Spoke

Central launchpad radiating outward to isolated spokes. Must return to hub.

Example: Smartwatch OS

Model 2

Fully Connected

Decentralized mesh where every node directly interconnects contextually.

Example: Wikipedia

Model 3

Multilevel / Tree

Strict parent-child hierarchical taxonomy with category branches and leaves.

Example: Amazon

Model 4

Stepwise (Linear)

Strict sequential progress (1 → 2 → 3 → 4). Peripheral nav stripped.

Example: Stripe Checkout

Model 5

Pyramid Model

Apex entry fanning out into tiered depth with lateral sibling traversal.

Example: Duolingo

Model 6

Flat Navigation

All sections sit on the exact same hierarchical tier. 1-click peer switching.

Example: Linear / IG

1. Hub and Spoke Navigation

Hub and Spoke: All navigation originates from a central **Hub** (home or master dashboard). Users navigate outward along independent **Spokes** (sub-pages or discrete tools). To move between Spokes, users must return to the central Hub.

Structural Anatomy

- **Central Home Anchor:** The Hub acts as the single source of truth, organizing disparate tasks.
- **Strict Task Isolation:** Activities within Spoke A cannot bleed into or disrupt Spoke B.
- **Backtracking Requirement:** Traversal between Spoke 1 and Spoke 2 requires an intermediate return (Spoke 1 → Hub → Spoke 2).
- **Modular Growth:** Adding new spokes does not require reorganizing other existing spokes.

Navigational Topology

Star Graph

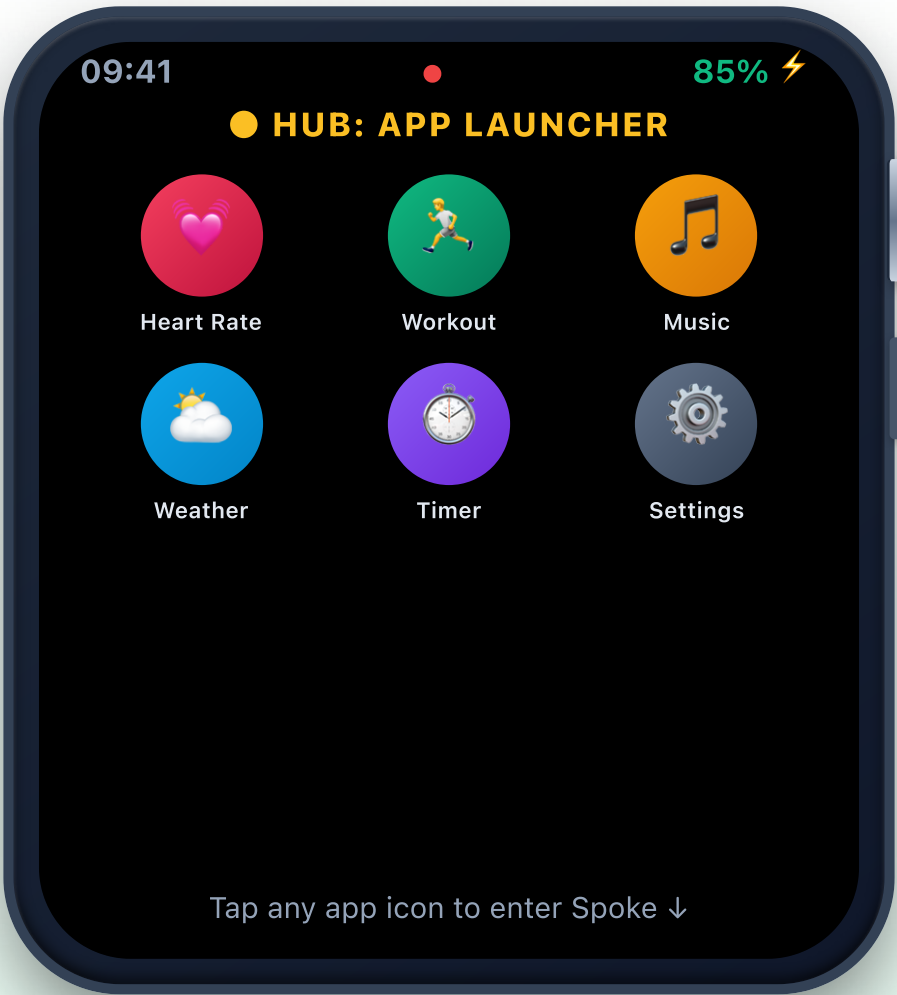


Independent sub-tasks · Must backtrack to Hub to switch destinations

- **Radial Routing:** Clean mental model ("I always know how to get back home").
- **Zero Cross-Bleed:** User is never confused by deep nested branching.

 **Next Slide: Live Interactive Smartwatch Demo** →

Real-World Example - Smartwatch Interfaces



Hub & Spoke - Evaluation & Applicability

✓ Advantages (Pros)

- **Zero Disorientation:** Users always have a clear mental anchor ("Just go back Home").
- **Task Containment:** Work in one spoke (e.g. paying a utility bill) doesn't interfere with another.
- **Independent Scalability:** New features/spokes can be added as new tiles on the Hub without redesigning the app.
- **Ideal for Multi-Tool Portals:** Well-suited for users who enter, perform a discrete job, and exit.

✗ Disadvantages (Cons)

- **High Backtracking Cost:** Frequent switching requires excessive clicks: Spoke A → Hub → Spoke B.
- **Bottleneck at the Hub:** As features multiply, the Hub dashboard risks becoming bloated and cluttered.
- **Inefficient for Cross-Task Comparison:** Comparing data between two spokes side-by-side is awkward.

📍 Where It Is Applicable

- **Fintech & Mobile Banking:** Send Money, Pay Bills, View Statement, Transfer.
- **Smartwatch & Automotive UI:** Apple Watch circular app grid; Tesla in-car dashboard.
- **Self-Service Kiosks & ATMs:** Cash withdrawal, Balance check, PIN change.
- **Account Settings Portals:** Google Account (myaccount.google.com).

2. Fully Connected (Matrix / Mesh) Navigation

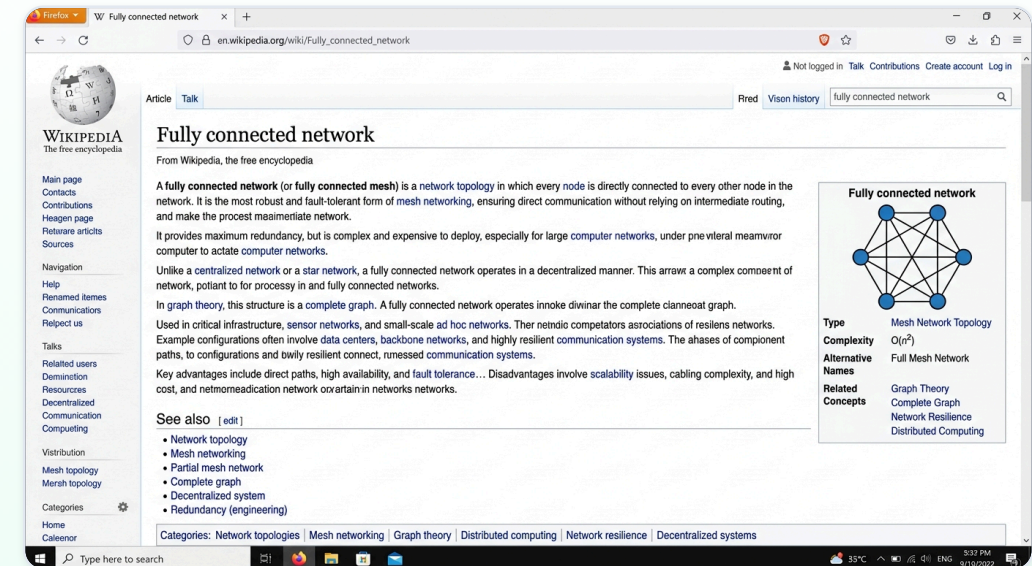
Fully Connected (Network / Mesh): Every node has direct, contextual connections to other nodes in the system without funneling through an intermediary hub or tree ancestor. Links are associative, organic, and multidirectional.

Structural Anatomy

- **Decentralized Hypertext:** No single mandatory starting node or fixed hierarchy.
- **Associative Pathways:** Navigation is triggered contextually by semantic relevance inside content.
- **Graph Topology (Kn):** In theory or practice, any topic can leap directly to any related concept.
- **Serendipitous Traversal:** Encourages exploratory learning and discovery rather than transactional task completion.

Real-World Example: Wikipedia

Encyclopedic
Mesh



Every paragraph contains contextual blue hyperlinks, infobox connections, and 'See also' cross-references.

Fully Connected · Evaluation & Applicability

✓ Advantages (Pros)

- **Maximum Exploration Freedom:** Users follow their own curiosity and mental associations without artificial roadblocks.
- **Shortest Cognitive Distance:** Jump straight to a related concept without navigating up to a category and back down.
- **Matches Human Cognition:** Closely mirrors the associative neural networks of human memory and conceptual mapping.

✗ Disadvantages (Cons)

- **"Lost in Hyperspace" Syndrome:** Users lose spatial awareness, forgetting where they started or how deep they have traveled.
- **No Definitive "Next Step":** Dangerous for transactional applications where users must complete a specific workflow.
- **Cognitive Choice Overload:** Dozens of competing blue links on a single screen trigger decision fatigue.

📍 Where It Is Applicable

- **Digital Encyclopedias & Wikis:** Wikipedia, Fandom, MDN Web Docs.
- **Personal Knowledge Management (PKM):** Notion bi-directional links, Obsidian graph views, Roam Research.
- **Infinite Canvases & Whiteboards:** Figma, Miro, MindMeister.
- **Entertainment & Media Databases:** IMDb, TV Tropes.

3. Multilevel or Tree (Hierarchical) Navigation

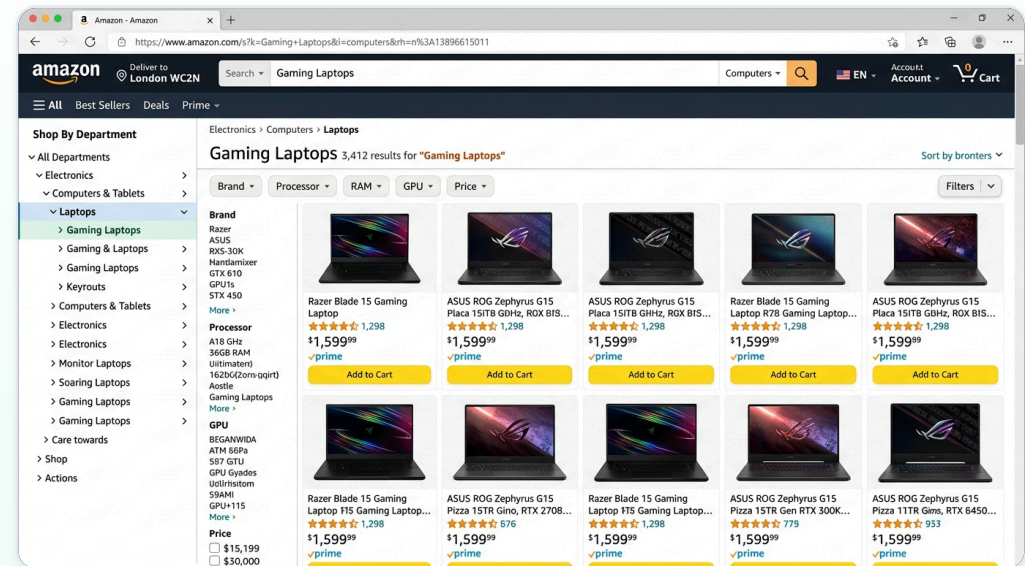
Multilevel / Tree Model: Information is organized into a top-down, parent-child taxonomy. Users start at a broad Root, drill down into Categories and Subcategories, and finally arrive at terminal Leaf nodes (e.g. specific product or article).

Structural Anatomy

- **Strict Ancestor-Descendant Lines:** Root → Department → Category → Subcategory → Product.
- **Wayfinding Signposts:** Uses *Breadcrumbs* (`Home > Electronics > Laptops`) and Mega Menus to clarify position.
- **Faceted Classification:** Allows users to prune branches using multi-attribute filters (Brand, Price, Specs).
- **Systematic Scalability:** Accommodates millions of SKUs without collapsing layout structure.

Real-World Example: Amazon.com

E-Commerce
Taxonomy



Left drawer shows deep branching: **All > Electronics > Computers > Laptops > Gaming Laptops.**

Multilevel Tree · Evaluation & Applicability

✓ Advantages (Pros)

- **Infinite Scalability:** Organizes hundreds of thousands of items cleanly without overwhelming the front page.
- **Familiar Mental Model:** Mimics real-world departmental stores (Aisle → Shelf → Product) and file systems.
- **Predictable Wayfinding:** Breadcrumb trails allow instant ascending back to any parent level in one click.
- **SEO & URL Friendly:** Creates clean, semantic directory structures (`/shop/electronics/laptops`).

✗ Disadvantages (Cons)

- **Pogo-Sticking Risk:** If a user chooses the wrong branch, they must climb back up and drill down another branch repeatedly.
- **Broad vs Deep Dilemma:** Too broad = overwhelming mega-menus; too deep = buried content that users never find.
- **Rigid Categorization:** Products that naturally fit in multiple categories (e.g. smartwatches as both tech and jewelry) create classification headaches.

📍 Where It Is Applicable

- **Large E-Commerce Portals:** Amazon, eBay, Daraz, Walmart, Best Buy.
- **Government & Civic Websites:** Gov.uk, Nepal Government national portals, municipal services.
- **Academic & University Libraries:** Departmental faculties, course catalogs, thesis repositories.
- **Operating System File Explorers:** macOS Finder, Windows File Explorer.

4. Stepwise Navigation (Linear / Wizard)

Stepwise (Linear / Wizard) Navigation: Guides users sequentially through a predetermined, step-by-step path (Step 1 → Step 2 → Step 3 → Completion). Peripheral navigation, menus, and sidebars are deliberately removed to focus attention on a single goal.

Structural Anatomy

- **Progress Tracker (Stepper):** Numbered indicators (e.g. 1 → 2 → 3 → 4) show total steps, completed steps, and remaining effort.
- **Directional Gating:** Forward movement requires satisfying input validation on the current step.
- **"Tunneling" Focus:** Eliminates headers, footer links, and marketing banners to minimize abandonment (Checkout Funnel).
- **Safe Backtracking:** Dedicated "Back" button preserves previously entered data safely.

Real-World Example: Checkout Wizard

Step 1: Customer Info Step 2: Shipping Address Step 3: Payment Method Step 4: Order Review

Step 2: Shipping Address

Full Name
Jane Doe

Street Address
123 Main St

Apartment/Suite (optional)

City
Springfield

State/Province
California

ZIP/Postal Code
90001

Country
United States

☐ Save this information for next time

[Return to Customer Info](#) [Continue to Payment](#)

Top stepper: 1. Info → 2. Shipping (Active) → 3. Payment → 4. Review. Clean, distraction-free card.

Stepwise Navigation - Evaluation & Applicability

✓ Advantages (Pros)

- **Reduces Cognitive Overload:** Breaks intimidating forms (40 fields) into digestible 4-field chunks (Miller's Law).
- **Drastically Lowers Error Rates:** Validates data at each milestone before allowing users to proceed.
- **High Task Completion Rates:** Clear progress feedback gives users psychological momentum to reach the finish line.
- **Distraction-Free:** Removing secondary links prevents users from abandoning critical revenue funnels.

✗ Disadvantages (Cons)

- **Zero Non-Linear Freedom:** Experienced users cannot skip ahead or fill fields out of order.
- **Frustrating to Edit Past Data:** Clicking "Back" multiple times to fix a typo in Step 1 feels cumbersome.
- **Wizard Fatigue:** If a flow extends beyond 5–7 steps without clear value, abandonment spikes dramatically.

🎯 Where It Is Applicable

- **E-Commerce Checkout:** Shopify, Stripe Checkout, Amazon Buy Now.
- **Complex Financial & Tax Filings:** TurboTax, Banking loan applications, KYC identity verification.
- **Travel & Flight Booking:** Skyscanner, Airbnb booking, Expedia seat selection.
- **Software Installation & Onboarding:** Multi-step app setup wizards.

5. Pyramid Navigation Model

Pyramid Navigation: Starts with a single big topic or milestone at the **top**, and expands into wider, detailed lessons as you go **down**. The key feature is that users can also **move sideways** between related lessons on the same level without going back to the start.

How It Works (Simple Anatomy)

- **1. Start at the Top (Big Picture):** You begin at a single main goal or unit (e.g., *Unit 1: "Order Food & Drinks"*).
- **2. Go Deeper for Details:** Moving downward reveals smaller, specific lessons (e.g., *Greetings* → *Cafe* → *Numbers* → *Menu*).
- **3. Move Sideways (No Dead Ends):** Unlike a strict tree, you can jump directly between lessons at the same level without climbing back to the top.
- **4. Unlock as You Learn:** Deeper, harder levels unlock only after you finish the basic lessons above.

Real-World Example: Duolingo Learning Tree

Gamified Path



Top: Unit 3 ("Order Food") → **Middle:** Greetings & Cafe → **Bottom:** Practice & Checkpoints. Students learn step-by-step and can switch between parallel lessons.

Pyramid Model - Evaluation & Applicability

✓ Advantages (Pros)

- **Great for Step-by-Step Learning:** Beginners master simple basics first before tackling difficult topics.
- **Never Feel Lost:** You always know the big goal at the top while working on small lessons.
- **Easy Sideways Navigation:** Jump between related topics on the same level without starting over.
- **Fun & Rewarding:** Unlocking new tiers motivates users to keep going (gamification).

✗ Disadvantages (Cons)

- **Hard to Fit on Small Screens:** Wide branches take up a lot of screen space on mobile phones.
- **Can Annoy Advanced Users:** Experienced users might get frustrated if forced to unlock basics first.
- **Hard to Change:** Editing or moving one lesson can affect all the unlocked levels below it.

🎯 Where to Use It

- **Language & Learning Apps:** Duolingo, Khan Academy, Codecademy course syllabi.
- **Video Game Skill Trees:** RPG character upgrades (unlocking powers level-by-level).
- **Online News & Articles:** Big headline and summary at the top → deep detailed sections below.
- **Employee Onboarding:** Training programs that teach company skills in structured stages.

6. Flat Navigation Model

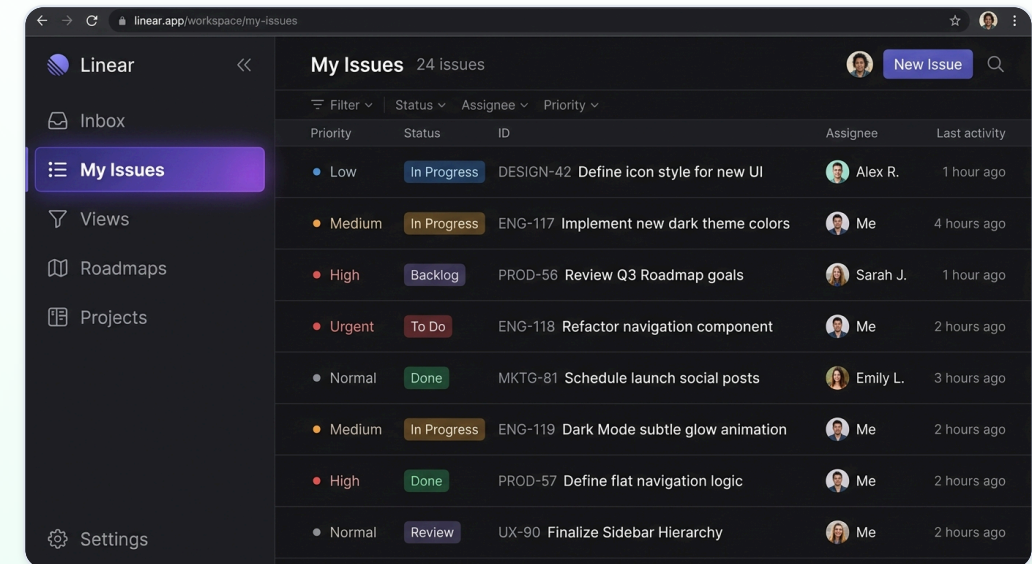
Flat Navigation: All primary sections or destinations exist on the **exact same hierarchical tier** with zero nesting. Every view is a direct peer, accessible with a single click or tap from anywhere in the interface.

Structural Anatomy

- **Zero Hierarchical Depth:** No sub-categories or child levels; every primary page is a sibling.
- **Persistent Global Dock / Bar:** Displayed via top tabs, a left icon rail, or a mobile bottom navigation bar.
- **Instant 1-Click Access:** Users can switch between views (e.g. Inbox → Issues) with 0 intermediate clicks and 0 back-stack traversal.
- **Equal Visual Weight:** Every destination has identical prominence and priority.

Real-World Example: Linear App

Single-Tier
Workspace



Sidebar presents peer views: **Inbox • My Issues • Views • Roadmaps • Projects • Settings.**

Flat Navigation · Evaluation & Applicability

✓ Advantages (Pros)

- **Fastest Interaction Speed:** Direct 1–tap/1–click switching to any core section of the app.
- **Zero Cognitive Navigation Debt:** Users never get trapped in deep ancestor sub–levels or forgotten sub–menus.
- **High Discoverability:** All major functional areas of the app are persistently visible at all times.
- **Ergonomic on Mobile:** Fits the thumb zone naturally when anchored in a 4–5 tab bottom navigation bar.

✗ Disadvantages (Cons)

- **Severely Limited Scalability:** Can only support 4 to 7 items before visual clutter and Hick's Law degrade usability.
- **Cannot Handle Deep Catalogs:** Inadequate for rich content taxonomies without embedding secondary navigation inside each tab.
- **Screenspace Overhead:** The persistent navigation bar or dock claims permanent screen real estate.

📍 Where It Is Applicable

- **Social Media Mobile Apps:** Instagram, X (Twitter), TikTok (Feed, Search, Post, Activity, Profile).
- **Developer & Productivity SaaS:** Linear, Discord channel list, Trello board view, Slack sidebar.
- **Single–Purpose Utilities:** Weather apps, Calculator, Clock (Alarm, Stopwatch, Timer).
- **Portfolio One–Pagers:** Direct section anchor links (About, Work, Contact).

Comprehensive Comparison Matrix

Summary of all 6 architectural models across core UI/UX dimensions:

Navigational Model	Topology Shape	User Freedom	Cognitive Load	Backtracking Cost	Primary Real-World Fit
Hub and Spoke	Centralized Star	Moderate (Isolated)	Low (Home anchor)	High (Must return to hub)	Spotify, Banking, ATMs, Watch OS
Fully Connected	Decentralized Mesh	Maximum (Freeform)	High (Disorientation)	None (Direct jumping)	Wikipedia, Notion, Mind Maps
Multilevel / Tree	Hierarchical Taxonomy	Structured Top-Down	Moderate (Breadcrumbs)	Moderate (Pogo-sticking)	Amazon, Gov.uk, File Explorers
Stepwise (Linear)	Strict 1-D Pipeline	Constrained (Gated)	Lowest (Chunked steps)	Strict (Step-by-step back)	Stripe Checkout, TurboTax, Onboarding
Pyramid Model	Top Overview → Expanding Tiers	Downwards + Sideways	Moderate (Clear path)	Low within same level	Duolingo, Video Games, Learning Courses
Flat Navigation	Horizontal Peer Row	Immediate 1-Click	Lowest (Zero nesting)	Zero (Instant switch)	Linear, Instagram, Trello, Mobile Tabs

UX Architect's Heuristic Rule: Choose *Stepwise* when error-prevention is paramount; *Flat* when speed among 4–5 peers is required; *Tree* when catalog size exceeds thousands; *Hub & Spoke* when distinct tasks must not cross-contaminate; and *Fully Connected* for open-ended knowledge discovery.

Interactive Navigational Topology Simulator

Click each model below to watch how users traverse each structural topology:

1. Hub & Spoke →

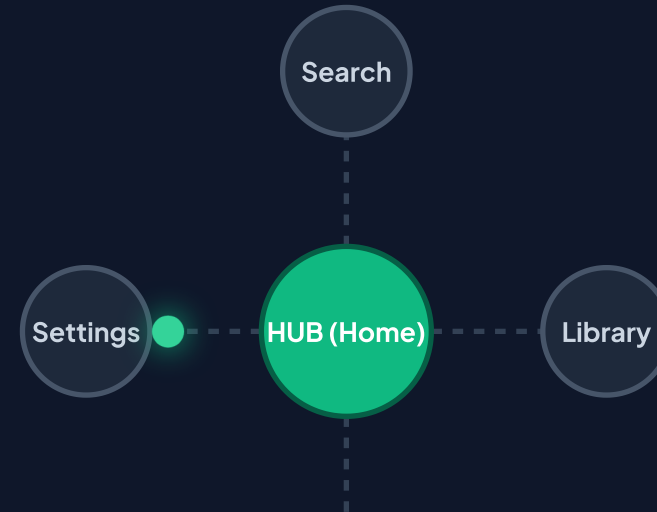
2. Fully Connected →

3. Multilevel Tree →

4. Stepwise (Linear) →

5. Pyramid Model →

6. Flat Navigation →



Hub & Spoke Model (Spotify / Mobile Banking)

Central Hub directs traffic outward. To move from Spoke A to Spoke B, the user MUST return to the central Hub. Prevents cross-contamination, but requires extra backtracking.

Section 5.4 Completed!

You have mastered all 6 core navigational models in Information Architecture: Hub & Spoke, Fully Connected, Multilevel Tree, Stepwise, Pyramid, and Flat Navigation.

Exam Quick–Recall Checklist (BCA 2nd Sem):

- **Hub & Spoke:** Central home, isolated tasks, high backtracking (Spotify / Banking).
- **Fully Connected:** Mesh graph, contextual hyperlinks, disorientation risk (Wikipedia).
- **Multilevel Tree:** Root → categories → leaves, breadcrumbs, high scalability (Amazon).
- **Stepwise:** Linear 1 → 2 → 3 → 4, distraction-free funnel, error reduction (Checkout).
- **Pyramid:** Starts at top overview, expands into detailed levels with easy sideways movement (Duolingo).
- **Flat:** Peer views, 1-click access, 4–5 tabs max, zero nesting (Linear / Instagram).