Building User Interfaces

Design Patterns and Design Languages

Professor Bilge Mutlu

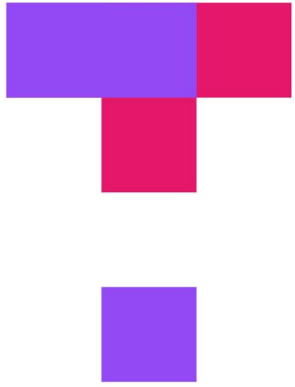
What we will learn today?

» Design Patterns

» Design Languages

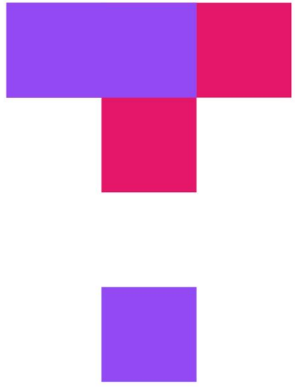
» Assignment Preview

TopHat Attendance



TOP HAT

TopHat Questions



TOP HAT

Design Patterns

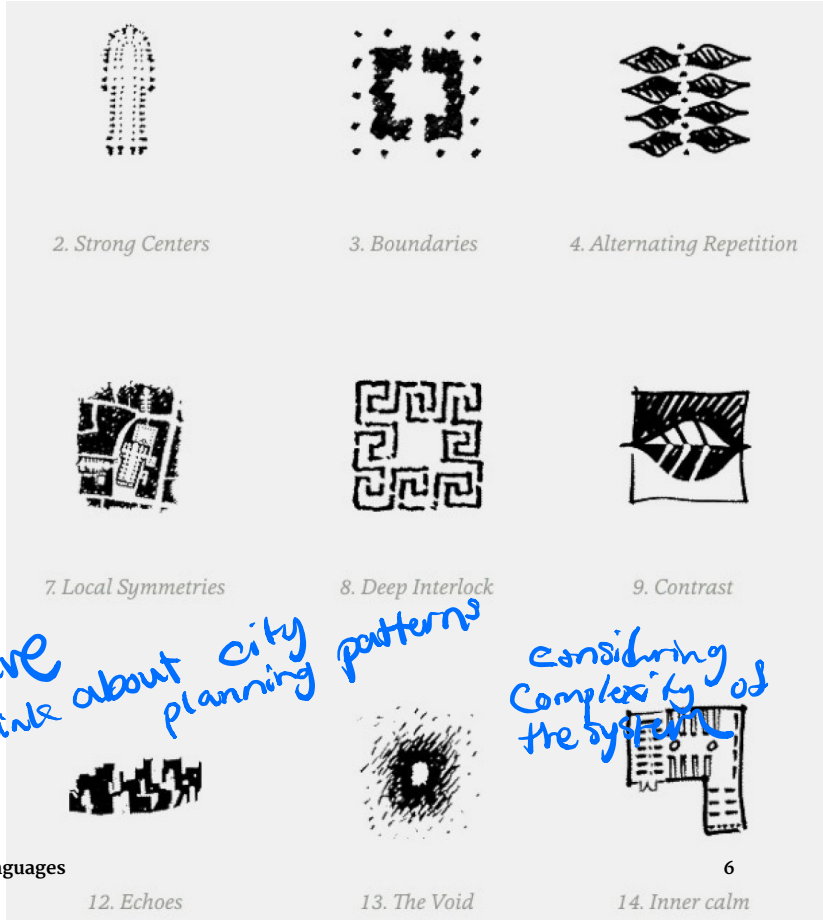
Recap: Design Patterns

Definition: A design pattern is a general, reusable solution to a commonly occurring problem within a given context.

Originally developed by Christopher Alexander (1977; *A Pattern Language*) to address problems in architecture and city planning.[1]

came from architecture - think about city planning patterns

considering complexity of the system



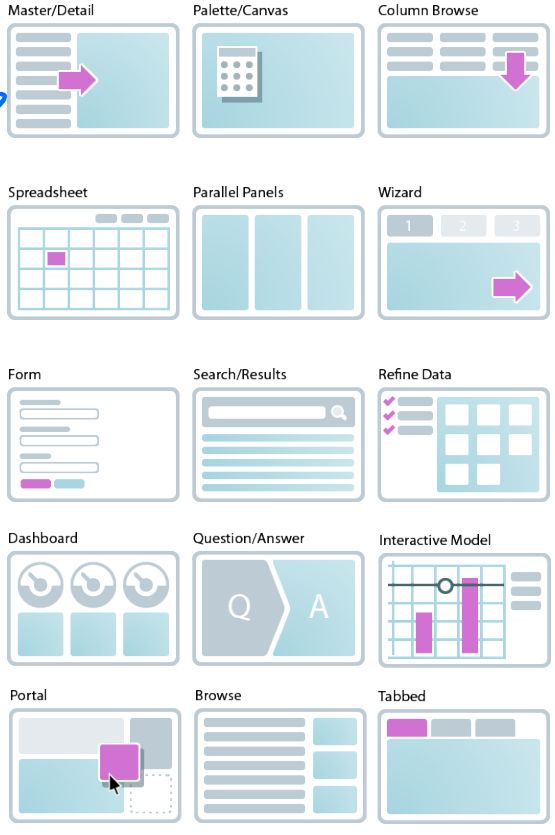
[1] Smart Cities Dive

Recap: Design Patterns in UX

In the last decade, designers have also developed and refined patterns for overall structure and organization, components and controls.[2]

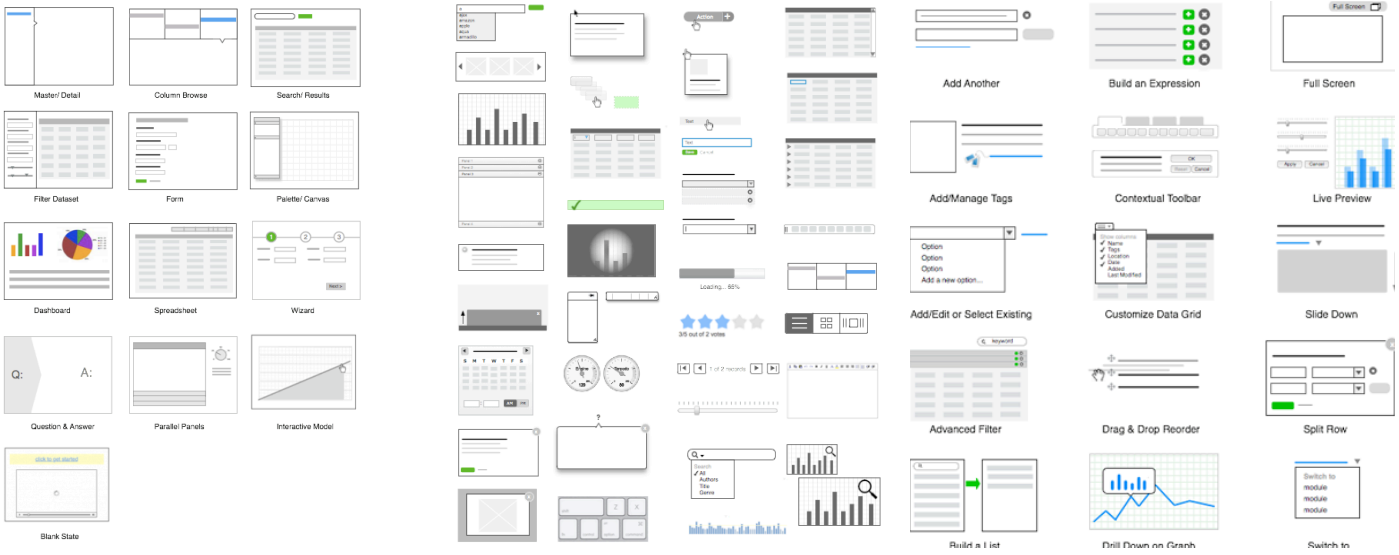
Problems tend to follow patterns → use a standardized set of solutions

common in email / message apps



[2] Neil, 2010, 12 Standard Screen Patterns

Source[3]



[3] Neil, 2010, 12 Standard Screen Patterns

The Problem with Patterns

Problem 1. Can I piece together different patterns to make a complete design? **No**, as this eclectic design would lack coherence.

Problem 2. How do I choose which pattern to use? Are patterns interchangeable? **No**, there has to be a *principle* to the selection of patterns.

Choices must be thoughtful!

Enter Pattern Languages

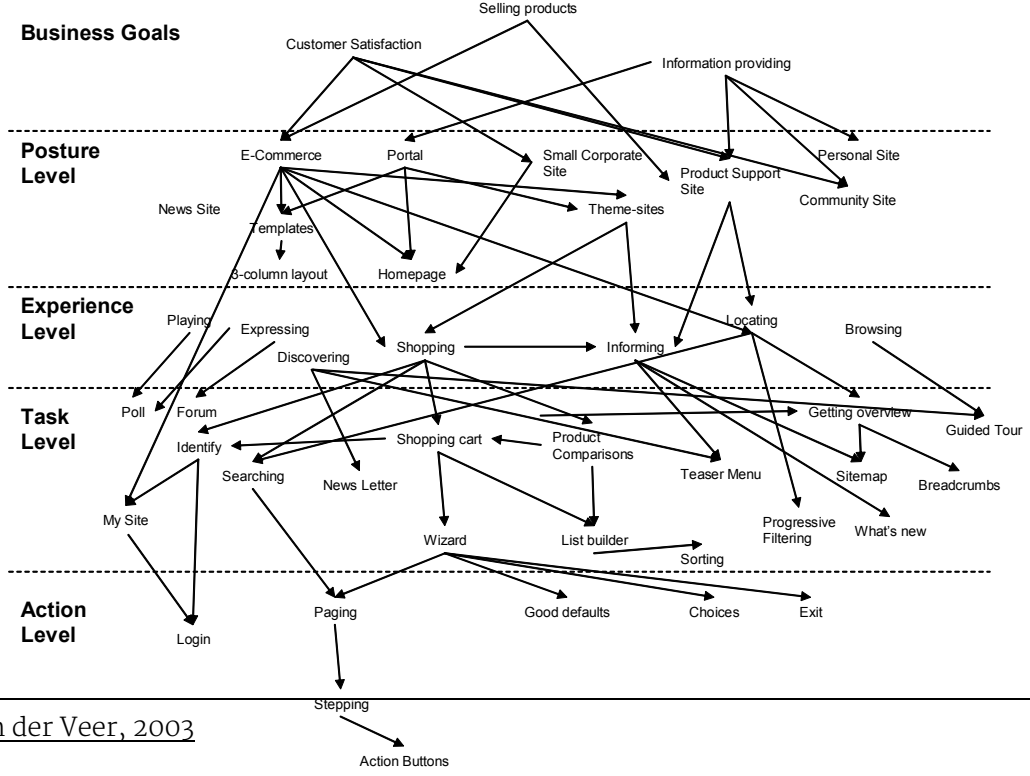
Define: A complete and hierarchical collection of patterns for a family of design problems.

Patterns are *words* (e.g., a component) that are connected with grammar rules to make *sentences* (e.g., a screen) and eventually *language* (e.g., user experience).[4]

The pattern language can be thought of as patterns being applied at different *levels*. Let's see an example.

[4] Kruschitz & Hitz, 2009

Source[5]



[5] van Welie & van der Veer, 2003

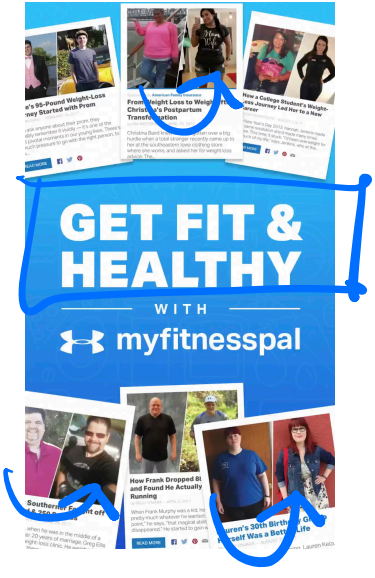
Business Goals

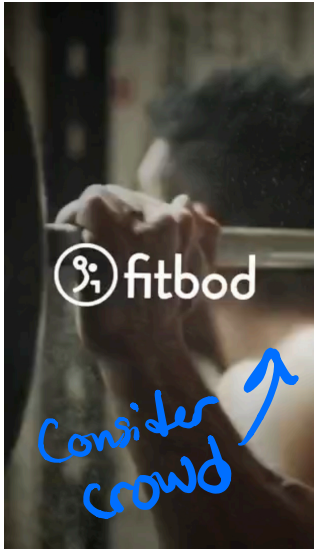
Definition: Conceptual design that captures the role that the design plays in user's life, i.e., the *mission* of the application, e.g., "helping users achieve fitness goals."

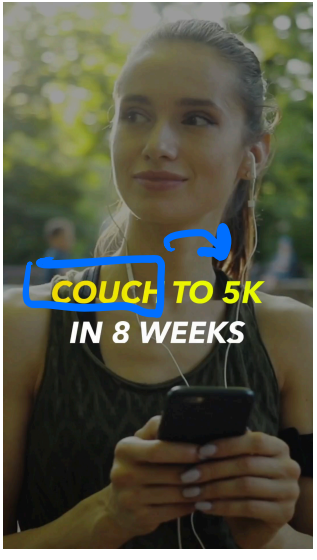
What do we want the user to be able to do?

Source[6]

Some examples







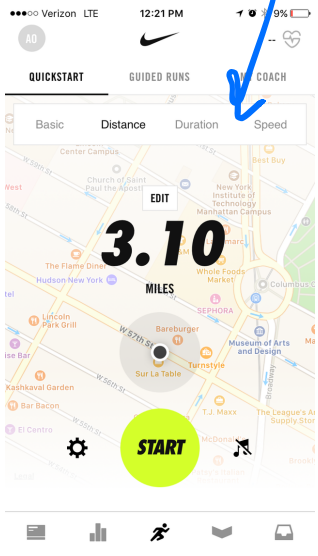
[6] Image sources (iOS App Store): *myfitnesspal*, *fitbod*, *5K runner*

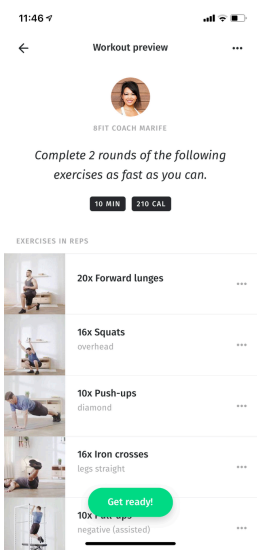
Posture-Level Patterns

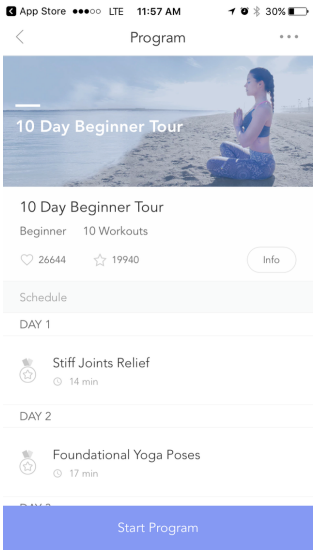
Definition: The *structure* that an application follows, i.e., what *type* of application it is, e.g., "a calorie tracking app," "a a step counter app," or "a life coaching app."

What kind of a system is this?

Source[7]







[7] Source for images

Elements of a Posture-level Pattern

Once we determine the posture of an application, it gives us guidance on:

» Structure

» Components

» User experience

» Alternatives/competitors

Structure: Central canvas with supporting panels[8]

Components: Canvas, dashboard, score panel, data summary

UX: Measurement during the activity, review later

Competitors: Strava, RunKeeper

[8] Image source

Experience-Level Patterns

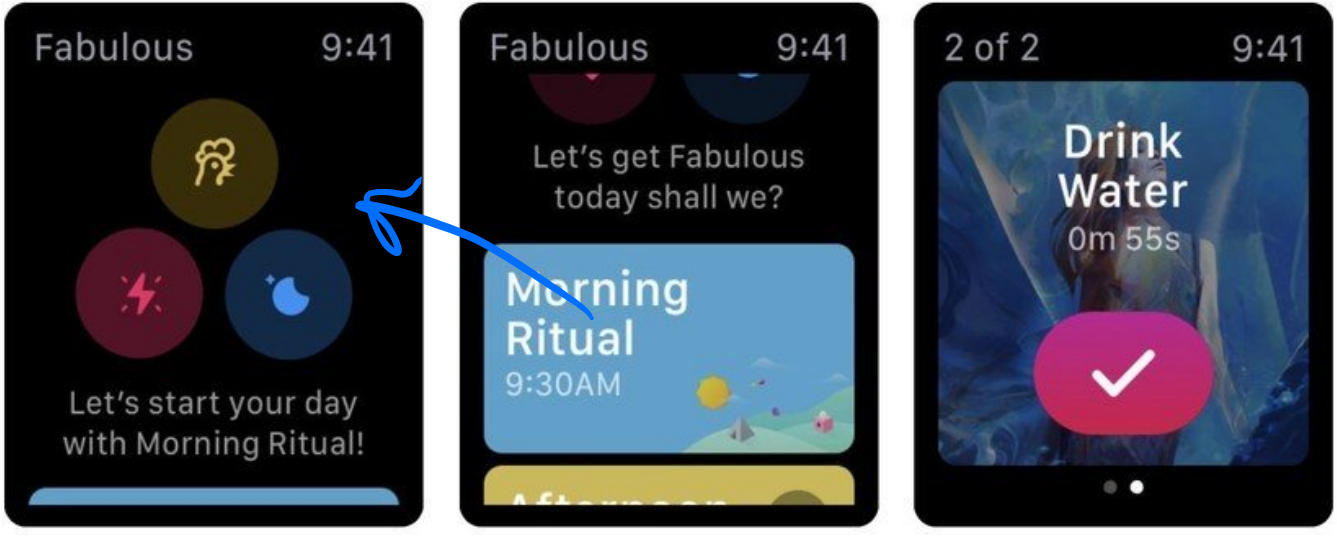
Definition: The *user goals* that make up the *user experience* that the application supports, e.g., activity tracking, coaching, and reviewing.

Experience-level patterns can also capture the *quality* of the user experience, e.g., *motivational* coaching.

what activities will they be doing?

Motivational app

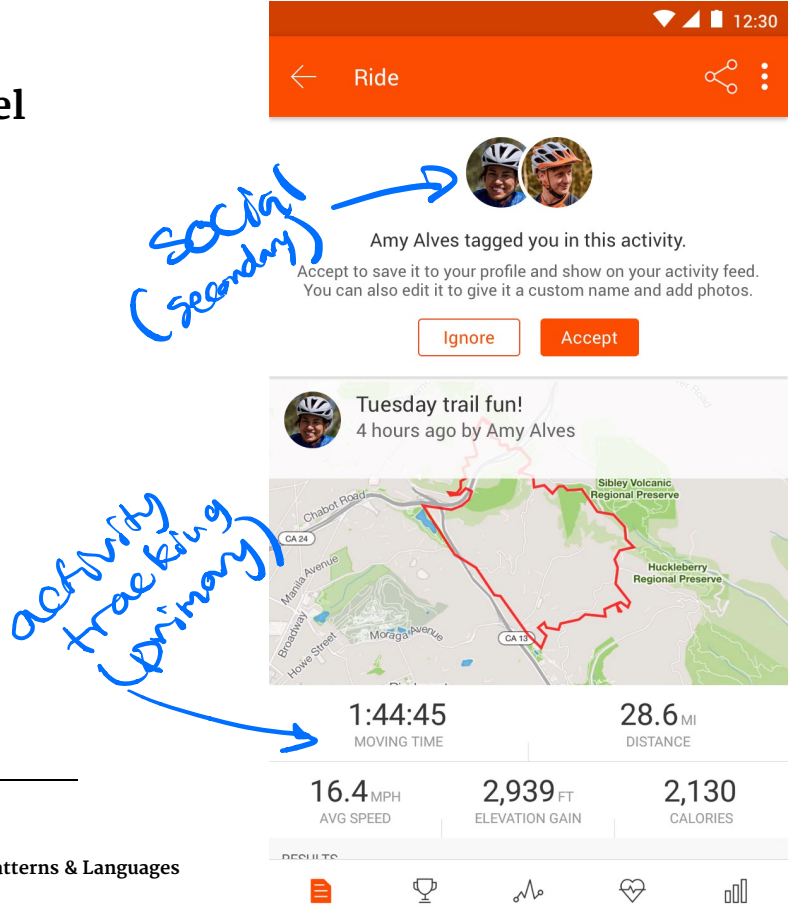
Source[9]



[9] Image source

Elements of an Exprience-Level Pattern[10]

- Primary goals, e.g., activity tracking
- Secondary goals, e.g., community building



[10] Image source

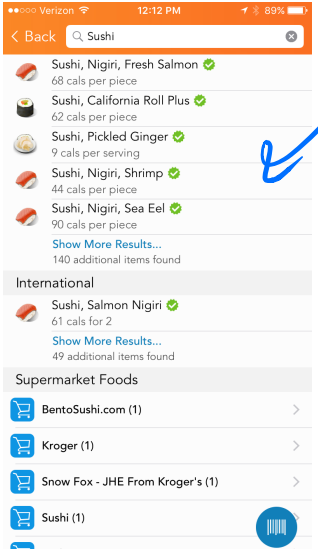
So far, mostly conceptual. This begins the more concrete levels

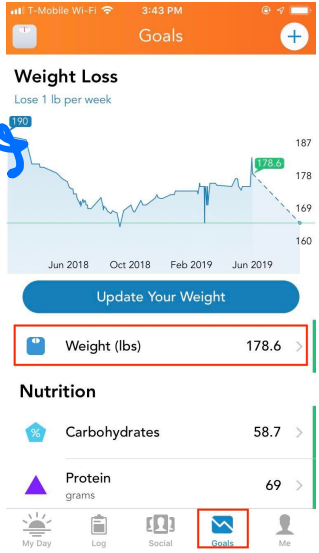
Task-Level Patterns

Definition: Design solutions that help users accomplish sequences of actions that make up user tasks, e.g., logging a meal, capturing a run, or completing a workout.

Tasks point to specific application *components*. E.g., meal logging can be done through a "search-and-filter" component, activity tracking can be done through a "scoreboard" component.

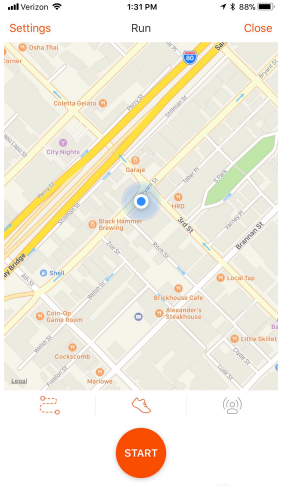
Source[11]



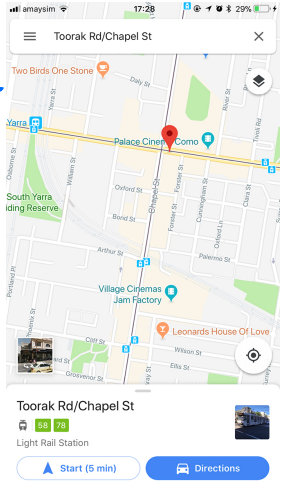


[11] Image sources: left, right

Task-level patterns can be domain independent. Business goals and posture-level patterns set the context for these patterns.[12]



Both use maps, but for a different purpose (set by more conceptual levels above)



[12] Image sources: left, right

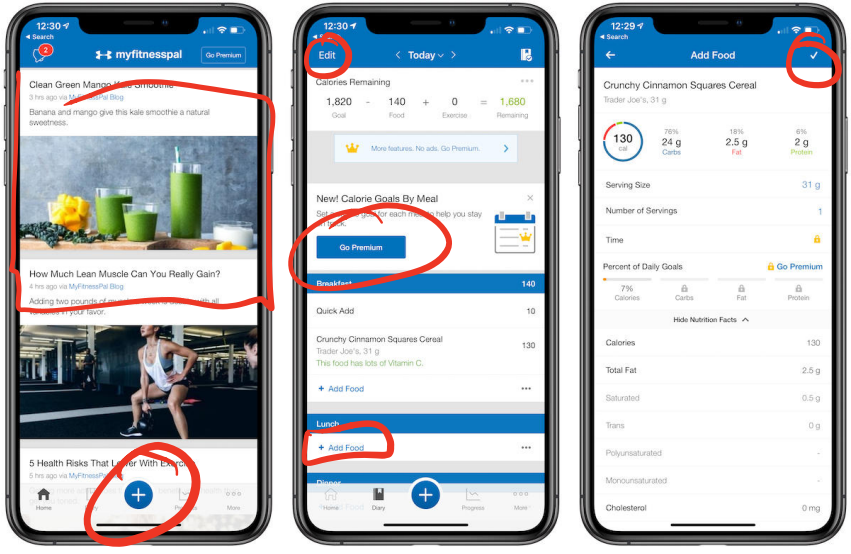
Action-Level Patterns

Definition: Design solutions that support the actions taken to complete the steps(s) of the user's task, e.g., a "start" button to initiate activity tracking, a selectable list entry for a food item.

Action-level patterns are the lowest level of building blocks for a design. They are often called *widgets* or *components* (as in React).

Low level components

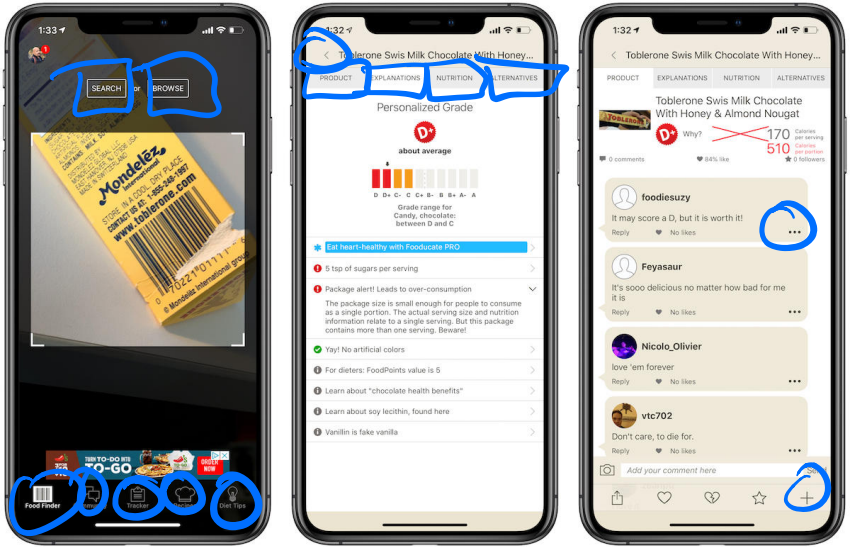
Action-level patterns for a *food tracking* app:[13]



Pretty much anything that can be interacted with can be an action-level pattern

[13] Image source: My Fitness Pal

Action-level patterns for a *food education* app:[14]

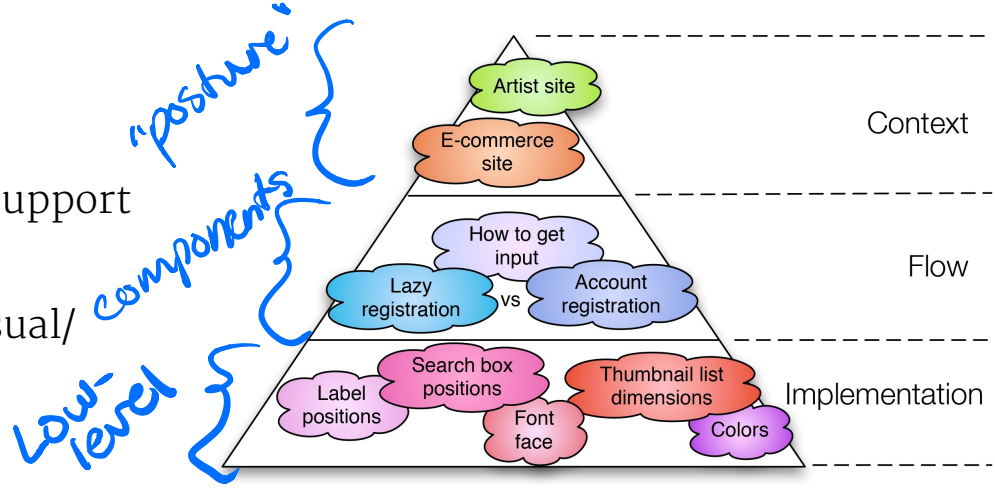


[14] Image source: Fooducate

A Simplifed Model[15] [16]

Three-levels of patterns:

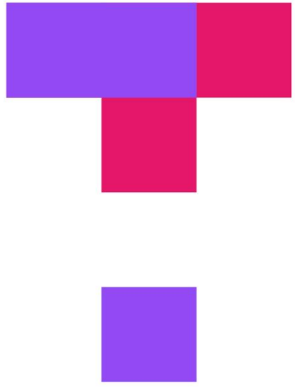
1. **Context:** Type of app
2. **Flow:** Components that support specific functions
3. **Implementation:** The visual/ behavioral elements that implement the functions



[15] Anders Toxboe

[16] More on the three-levels of patterns by Jerry Cao

TopHat Quiz



TOP HAT

How do we use pattern languages?

Common practice: Patterns in the higher levels are defined informally, and the task- and action-level patterns are adopted through experimentation and trial and error.

The problem: Ineffective (e.g., lack of coherence across different levels) and inefficient (wasted effort in experimentation).

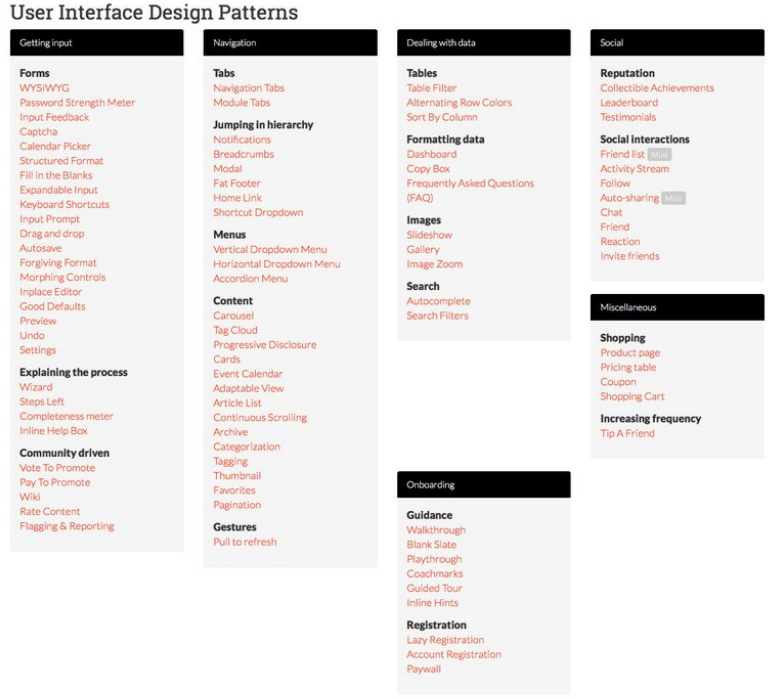
The solution: Defining patterns top to bottom will "generate" the design when patterns are available across all levels.[5]

---

[5] van Welie & van der Veer, 2003

Where do we find patterns?[17]

Task- and action-level patterns are organized into catalogues/ collections based on functional similarity.



[17] Image source

Online Pattern Libraries

- UIPatterns.io
- UI-Patterns
- Mobbin
- UI Garage
- Welie

Feel free to browse

We will not be going over creating patterns in this class

In-Class Activity

Source[18]

[18] Image sources: left, right

Business Goals
Mission of the application

Posture Level
"Type" of application

Experience Level
User goals

Task Level
Task sequences

Action Level
User actions

Help people track food consumption

Serving special nutritional needs

Track activity

Fitness Tracking

Healthy lifestyle

not exceed a certain amount of 'x'

make sure you eat a certain amount of "x"

I walked 'x' amount today

review trends

enter food

enter personal info

review current day

enter exercise

add "x"

show entered food

show entered miles/ exercise

go to another date

Design Languages

The problem: Pattern languages help you create a design that is consistent vertically. How do we create a system that is consistent *horizontally*? I.e., how do we achieve visual and behavioral consistency in designs?

The solution: Design languages!

Recap: Design Languages

Definition: A vocabulary of design elements that are repeatedly applied to interaction design problems.

Non-digital example: NASA Graphics Standard Manual.[19]

Nasa build a book to articulate vocabulary

[19] NASA

Source[19]

[19] NASA

UX Design Languages

Definition: Task- and action-level interface components that follow a consistent look and feel in appearance and behavior.

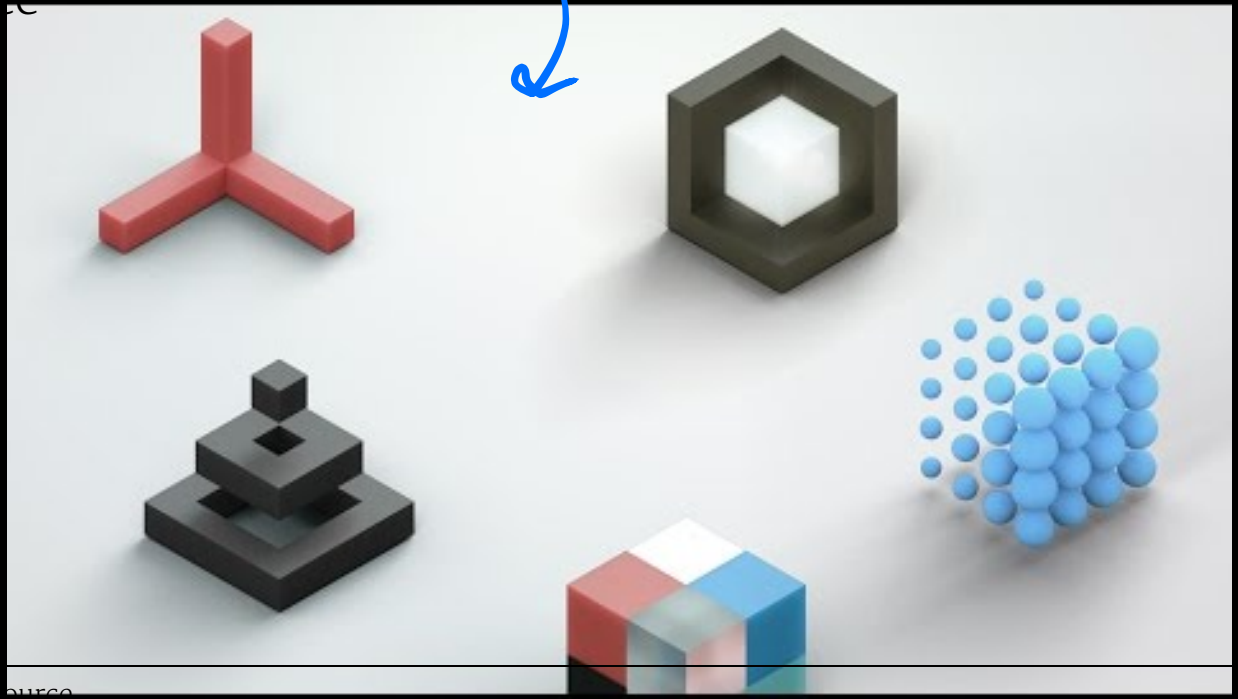
Commonly Used Design Languages[20]

- Material Design ← commonly used
- Fluent Design System ← microsoft
- Materialize ← beefier material
- Ant Design ← bit more subdued
- Grommet
- Flat Remix

} simpler options

[20] Image source

Fluent



Case Studies of Design Language Use

» Material studies examples

» Fluent design case studies

Assignment Preview

Design Assignment 07: Design Patterns

The assignment will help us generate design ideas for the React Native module deliverable: a *mobile fitness/calorie-tracking* application. In a two-part assignment, you will:

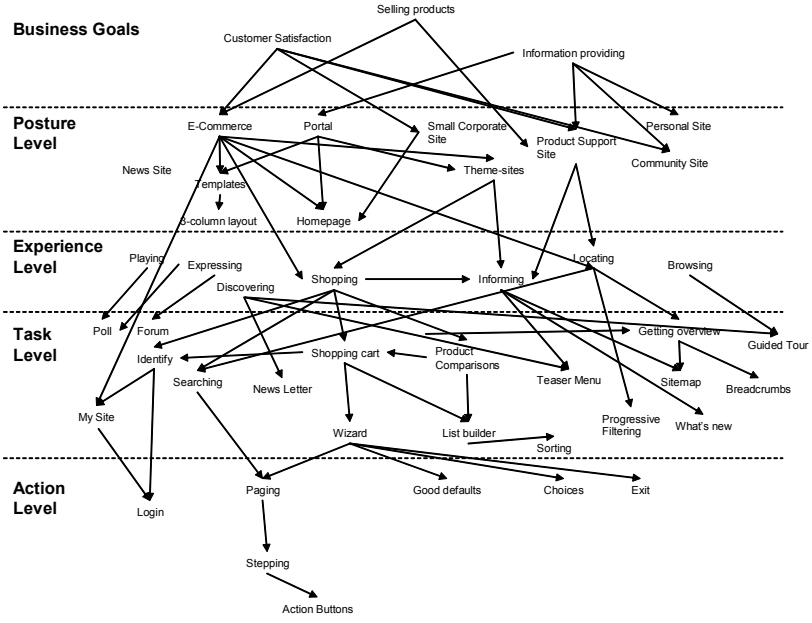
Step 1. Analyze the design patterns in an existing application

Step 2. Generate a design for a new application

Step 1. Analyze an existing design

» Analyze the application for all levels of design patterns, from top to bottom.[5]

» Analyze screen designs for task- and action-level patterns.

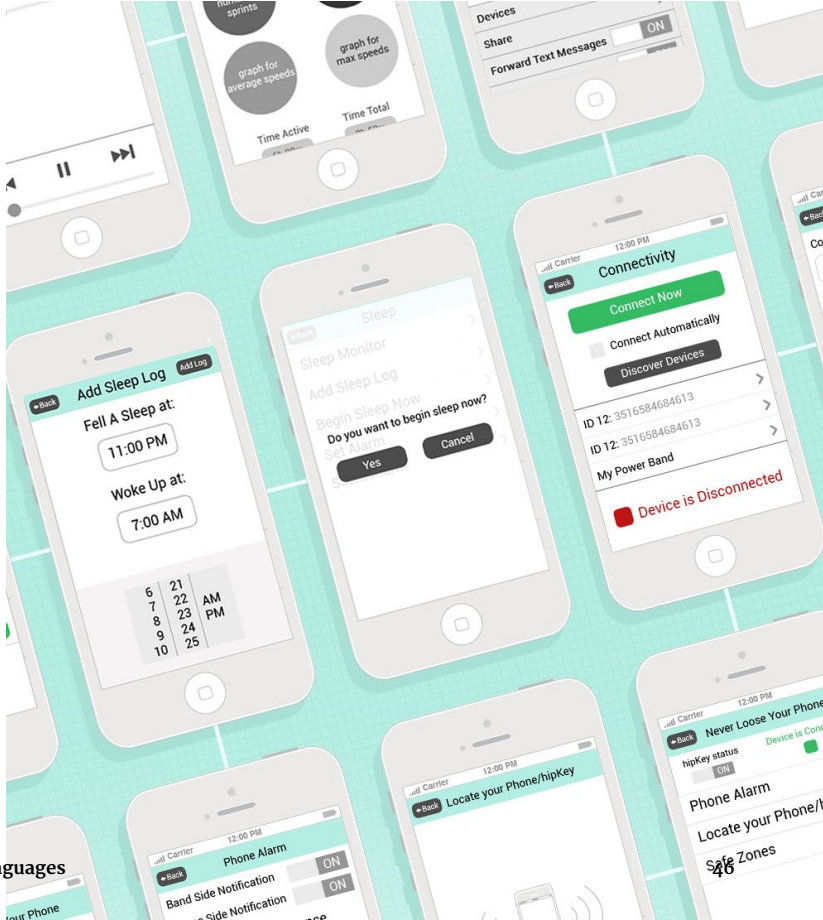


[5] van Welie & van der Veer, 2003

Step 2. Generate a new design[22]

» Using the pattern language analysis approach, make design decisions at all levels of your application, from top to bottom.

» Visualize your decisions at the task- and action-levels by creating wireframes.



[22] Image source

What did we learn today?

» Design Patterns

» Design Languages

» Assignment Preview