Designing Mobile Applications

Lecture 6

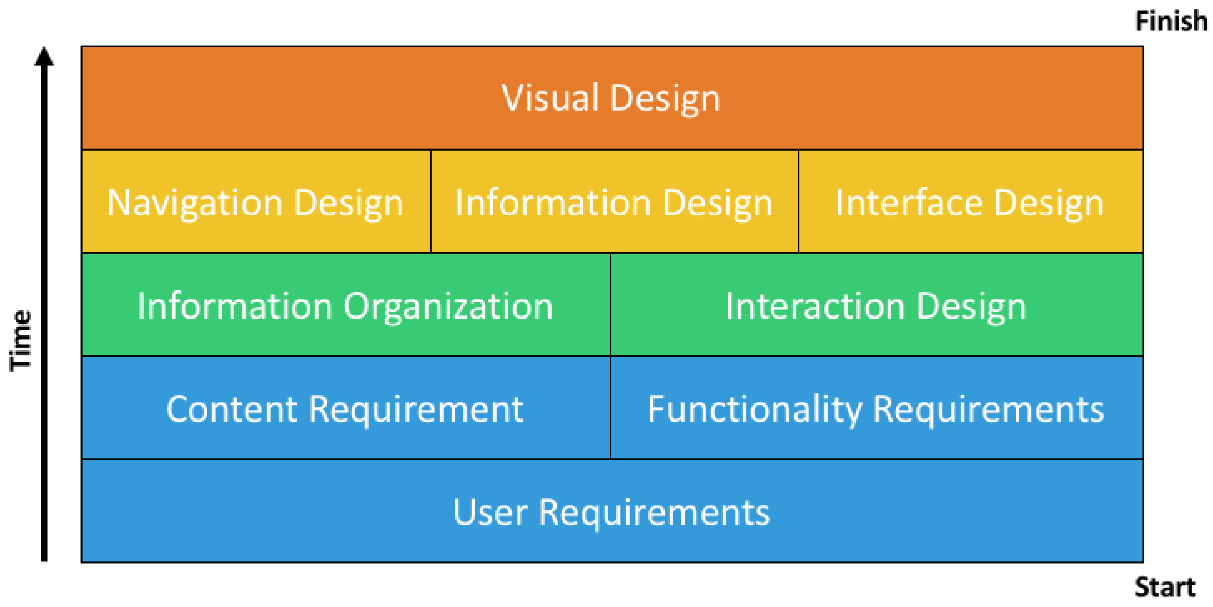
Mobile Computing

Slides by Dr. Tami Meredith

Information Architecture Stages

(after you have ALL your requirements)

- **Site Maps** → Information Organization
- **Storyboards** → Interaction Design
- **Wireframes** → Information and Navigation Design
- **Prototypes** → Interface Design
- **Implementation** → Visual Design



Finish

Visual Design

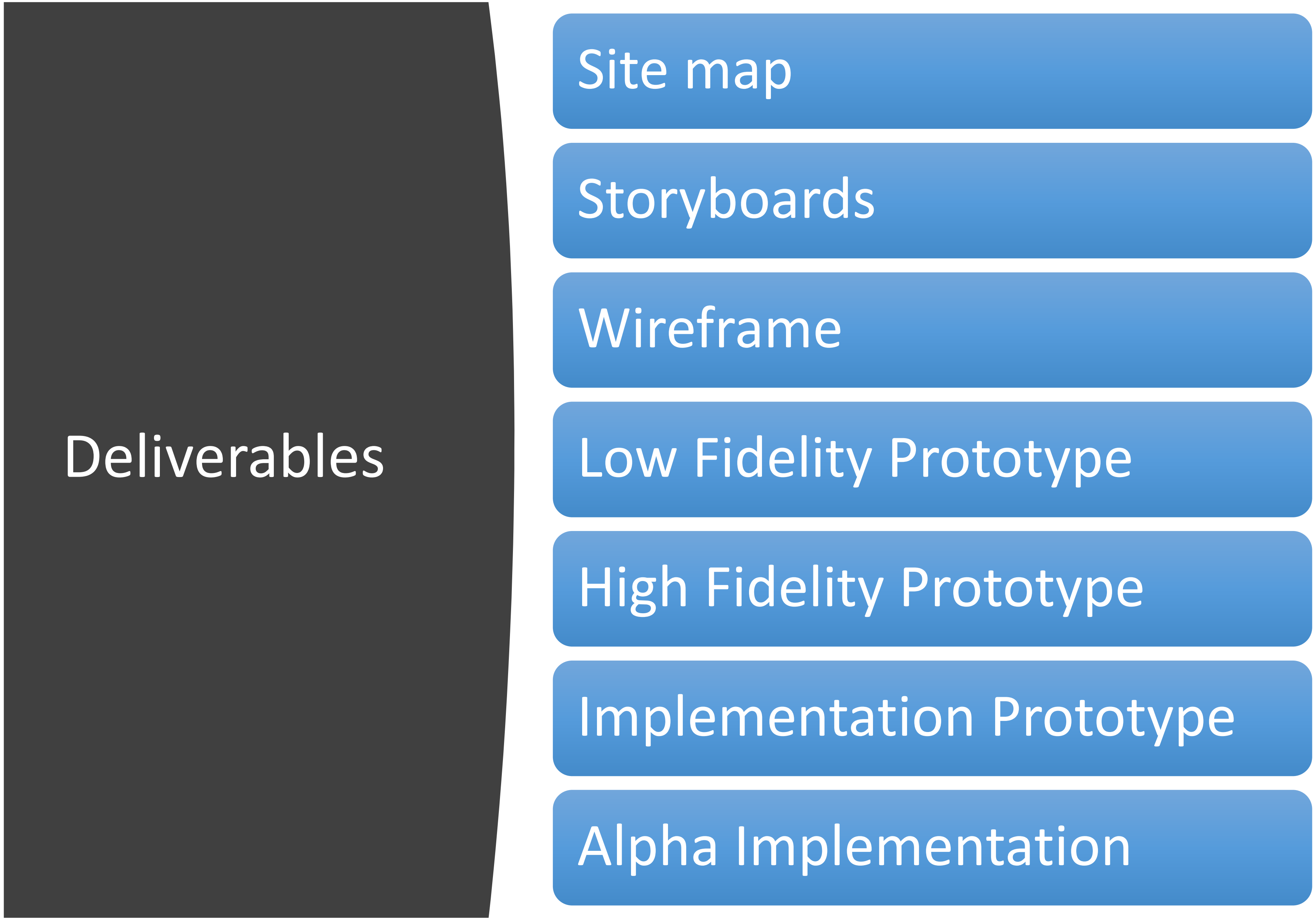
Navigation Design | Information Design | Interface Design

Information Organization | Interaction Design

Content Requirement | Functionality Requirements

User Requirements

Start

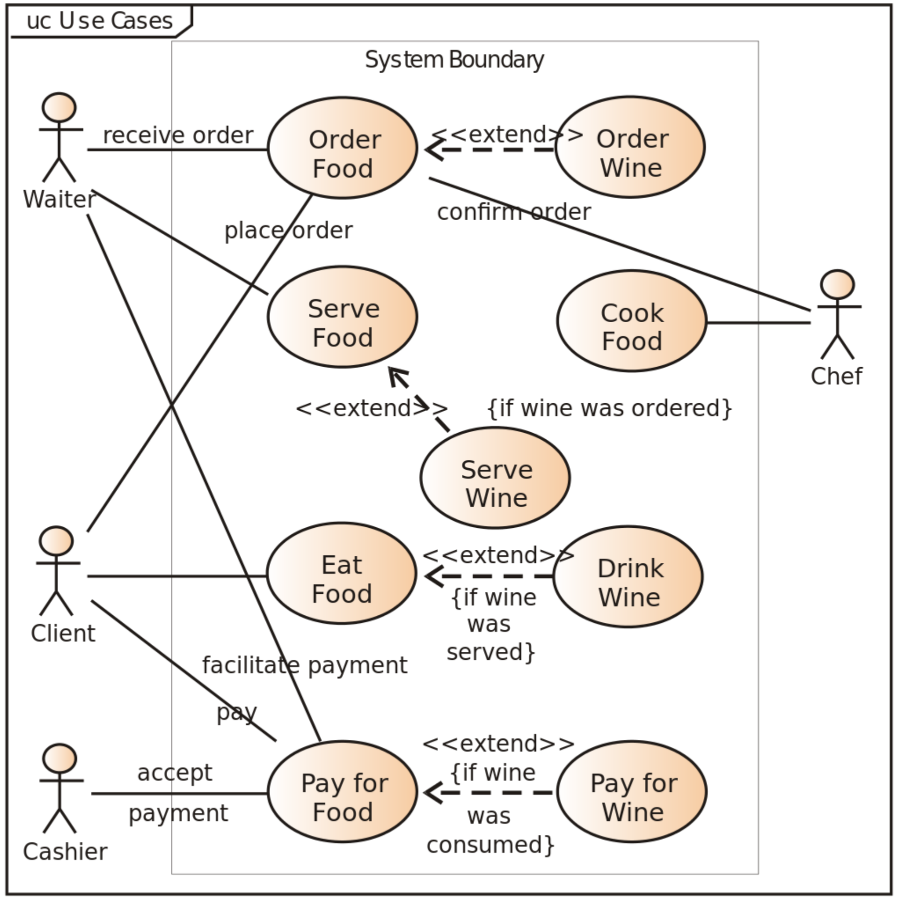


Deliverables

- Site map
- Storyboards
- Wireframe
- Low Fidelity Prototype
- High Fidelity Prototype
- Implementation Prototype
- Alpha Implementation

Use-Cases (In Software Requirements Specifications)

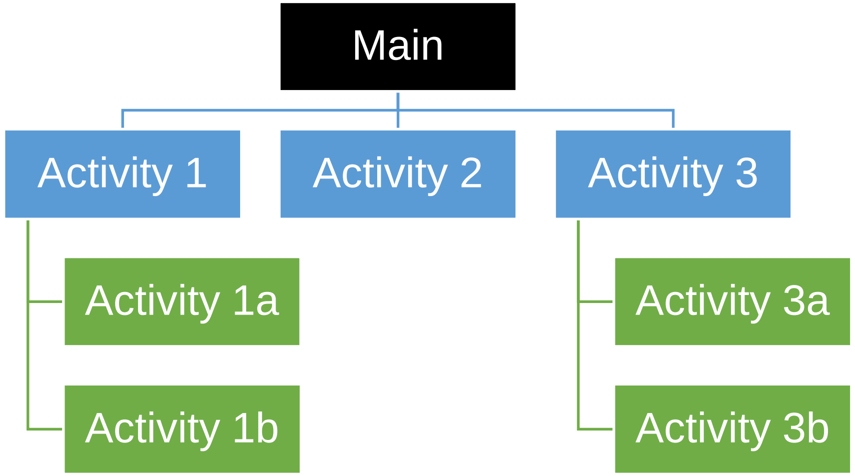
- A use-case is a list of actions or event steps typically defining the interactions between a role (known in UML as an actor) and a system to achieve a goal.
- The actor can be a human or other external system.
- It is a diagram showing the interaction between a user doing a task and the software.
- Use-Cases can really help you design your UI/UX.



Sitemaps

Site Maps

- Shows views and linked relationships (variant of an ER diagram)
- Create a **hierarchy** of screens, views, and/or functions.
- Root of hierarchy is main screen.
- Each child node should be a separate unit or group of functionality.



Main

Activity 1

Activity 2

Activity 3

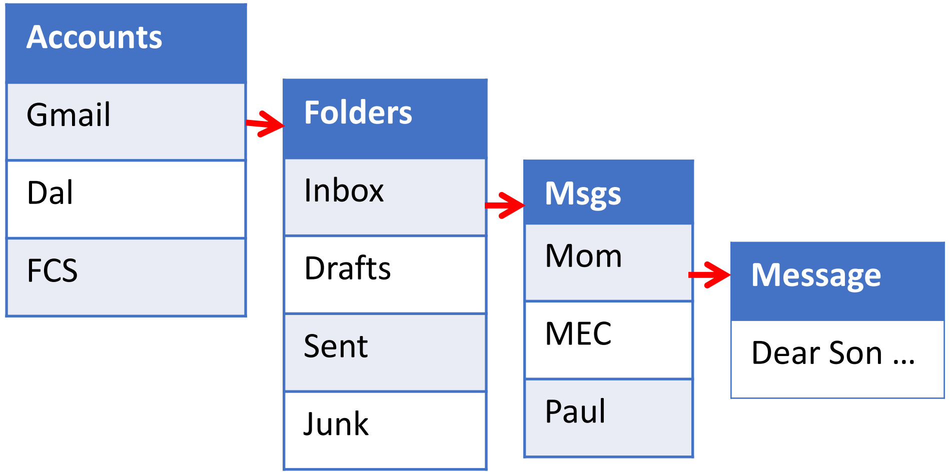
Activity 1a

Activity 1b

Activity 3a

Activity 3b

- Example: Outlook Mail app
  - List of accounts
  - List of folders in account
  - List of messages in folder
  - Message



Accounts

Gmail

Dal

FCS

Folders

Inbox

Drafts

Sent

Junk

Msgs

Mom

MEC

Paul

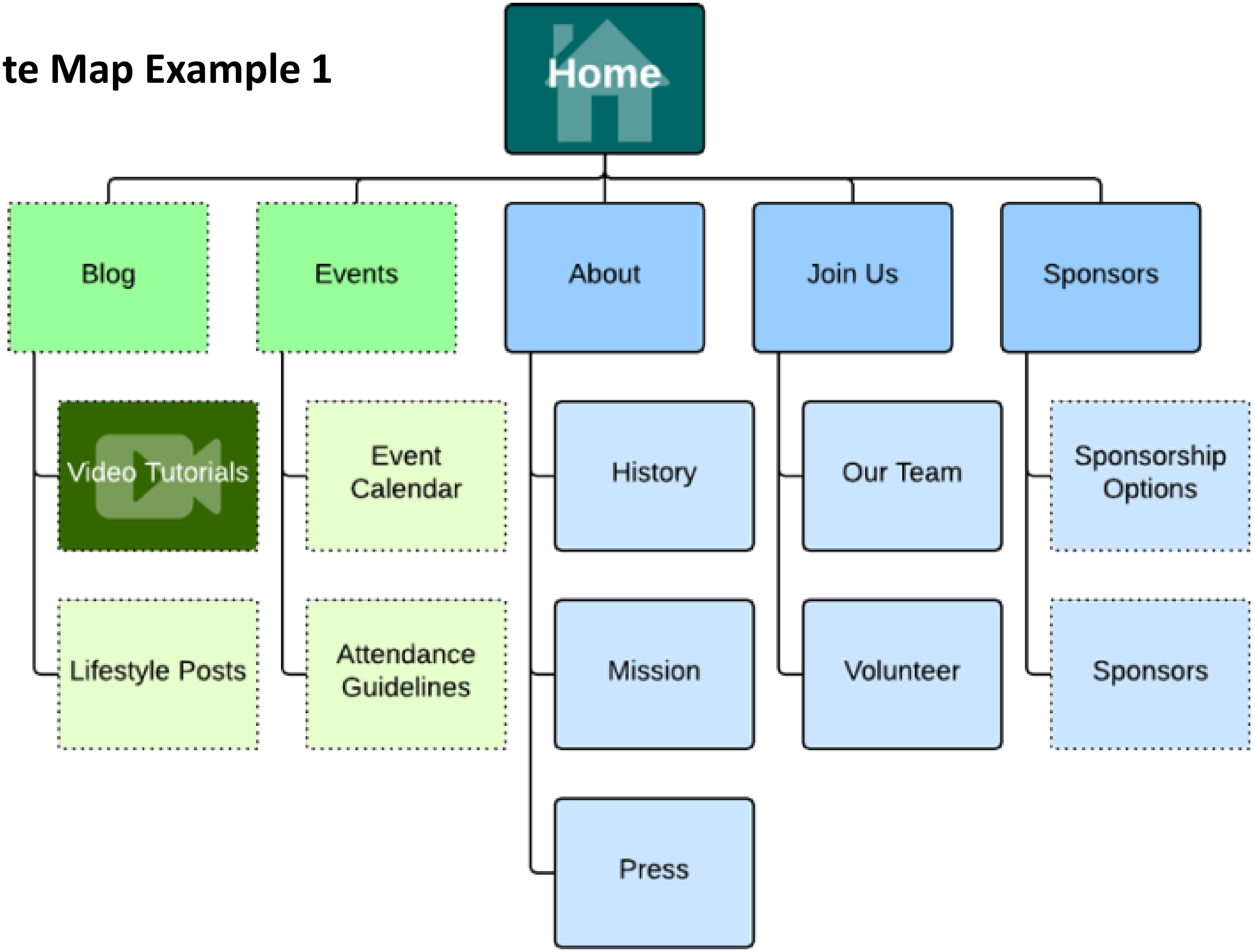
Message

Dear Son …

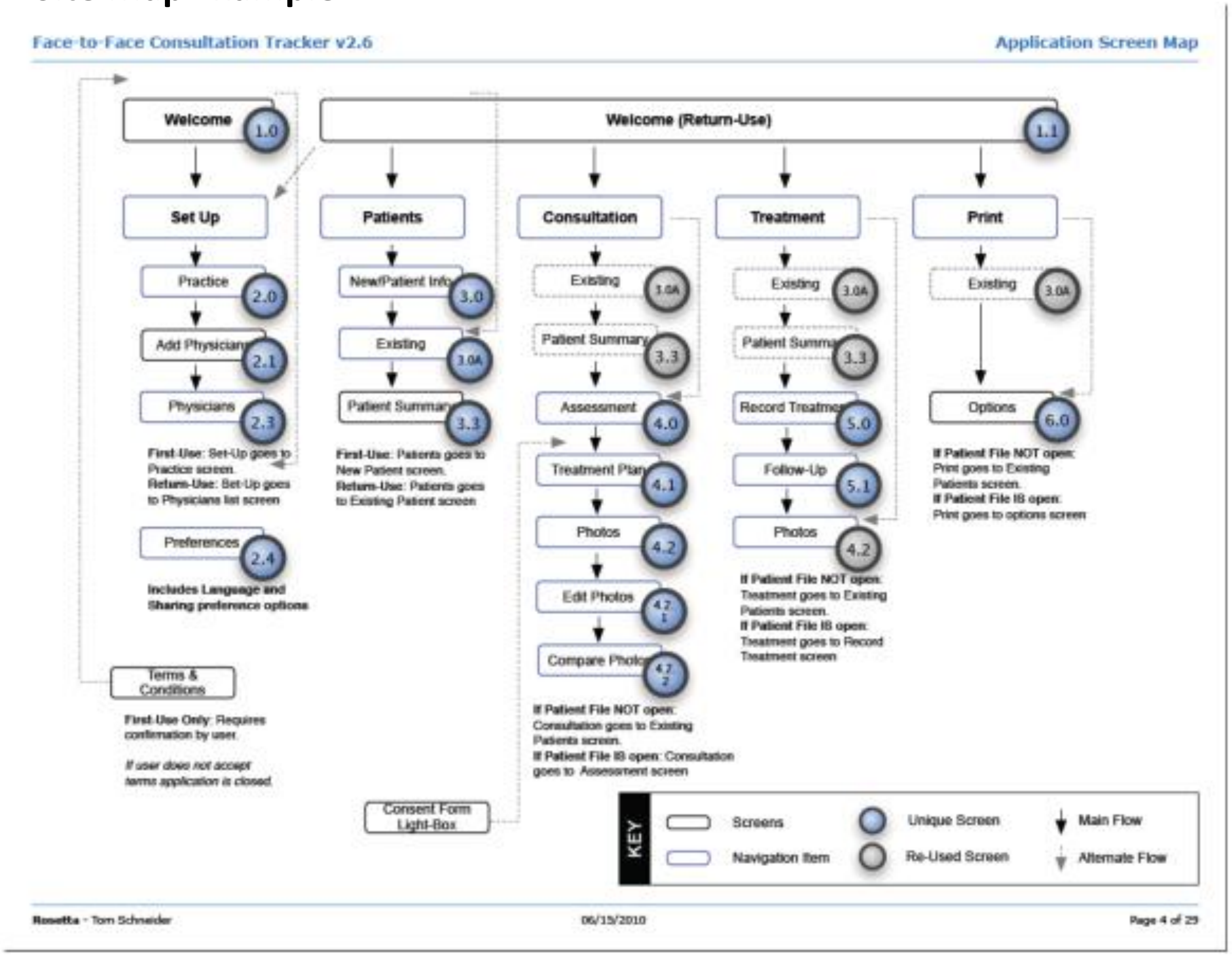
Guidelines for Site Maps

- Shallowness
  - Ideally no more than 5 levels of depth (i.e. columns).
    - People get lost easily in deep hierarchies.
- Low Branching
  - Limited number of items per view (few children).
    - Not enough screen space for long lists of children.
  - There is a trade-off between shallowness and low branching.
    - In many cases low branching is not possible (shallowness is usually more important).
- Clear labels on all children.
- Note: This does not apply to single-view applications. (e.g., iOS weather app)

Site Map Example 1



Site Map Example 2



Face-to-Face Consultation Tracker v2.6

Application Screen Map

First-Use: Set-Up goes to Practice screen.
Return-Use: Set-Up goes to Physicians list screen

Includes Language and Sharing preference options

First-Use Only: Requires confirmation by user.

If user does not accept terms application is closed.

First-Use: Patients goes to New Patient screen.
Return-Use: Patients goes to Existing Patient screen

If Patient File NOT open: Consultation goes to Existing Patients screen.
If Patient File IS open: Consultation goes to Assessment screen

If Patient File NOT open: Treatment goes to Existing Patients screen.
If Patient File IS open: Treatment goes to Record Treatment screen

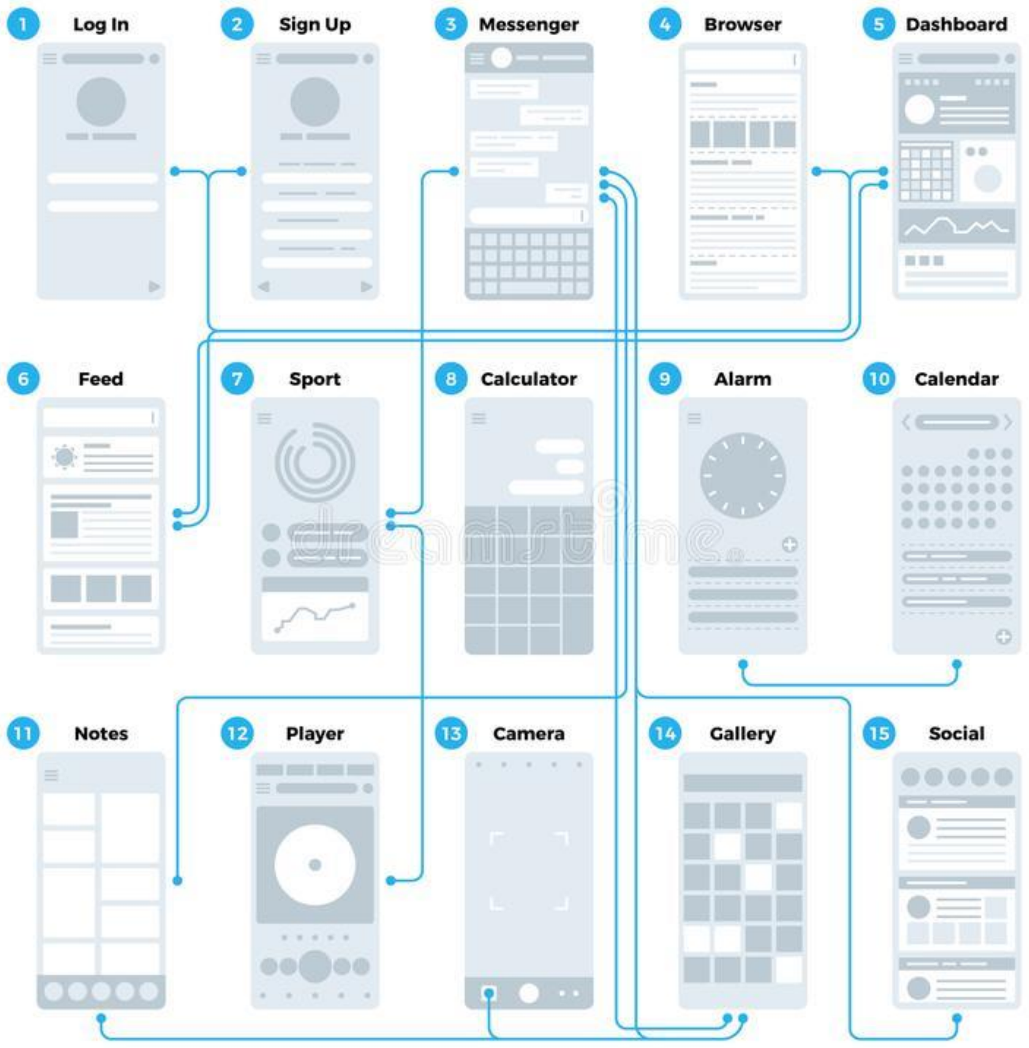
If Patient File NOT open: Print goes to Existing Patients screen.
If Patient File IS open: Print goes to options screen

Rosetta - Tom Schneider

06/15/2010

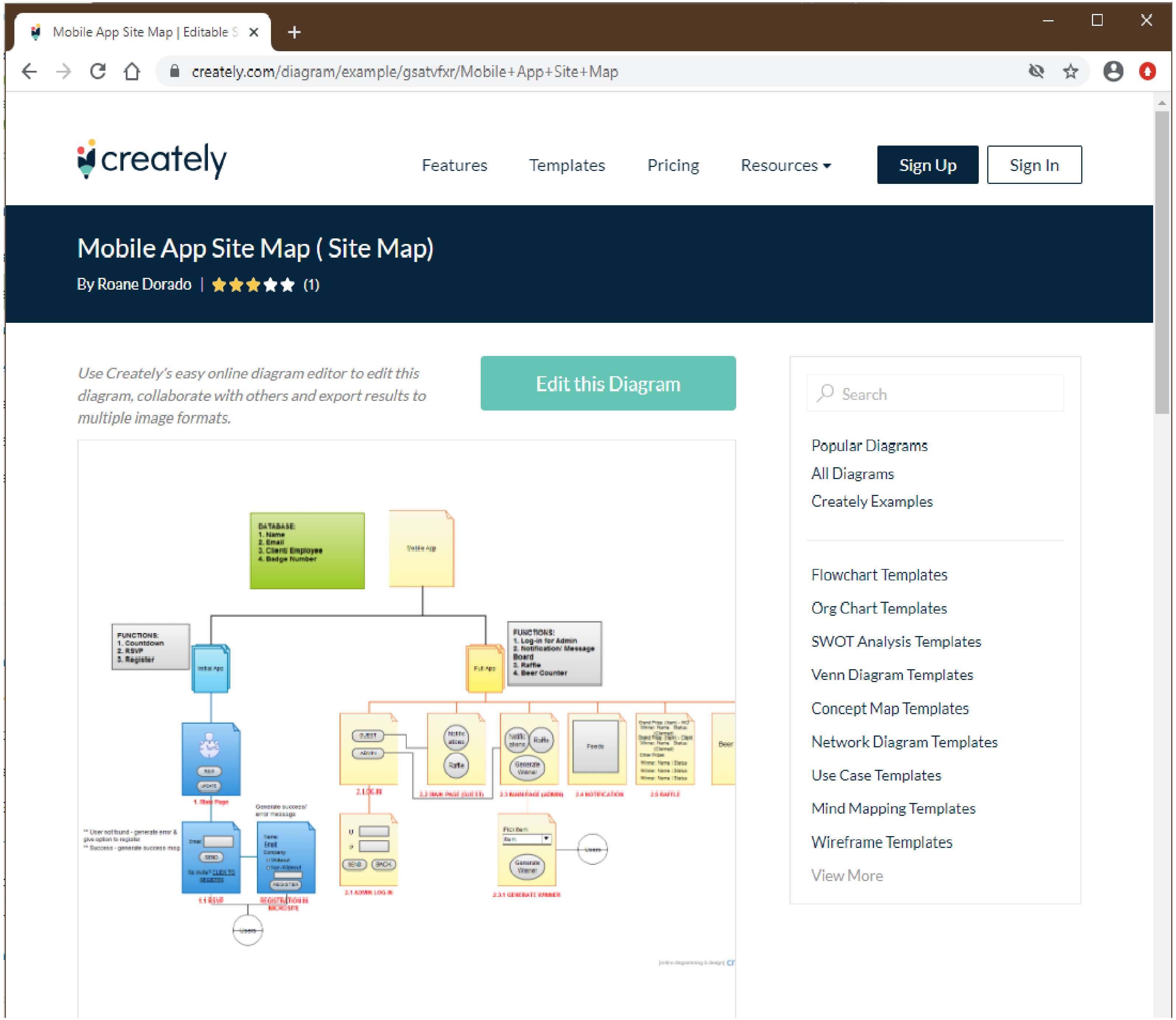
Page 4 of 29

Sitemap Example 3



7 Key Features of a Good Sitemap

1. High-level application features.
2. Required screens (including dialogues and popups).
3. View paths.
4. View functionality (what the view accomplishes).
5. Desired user behaviour.
6. User flow and movement.
7. Cruxes/bottlenecks: Potential areas for problems and complexity.



Mobile App Site Map | Editable S
creately.com/diagram/example/gsatvfxr/Mobile+App+Site+Map
creately
Features
Templates
Pricing
Resources
Sign Up
Sign In

Mobile App Site Map (Site Map)

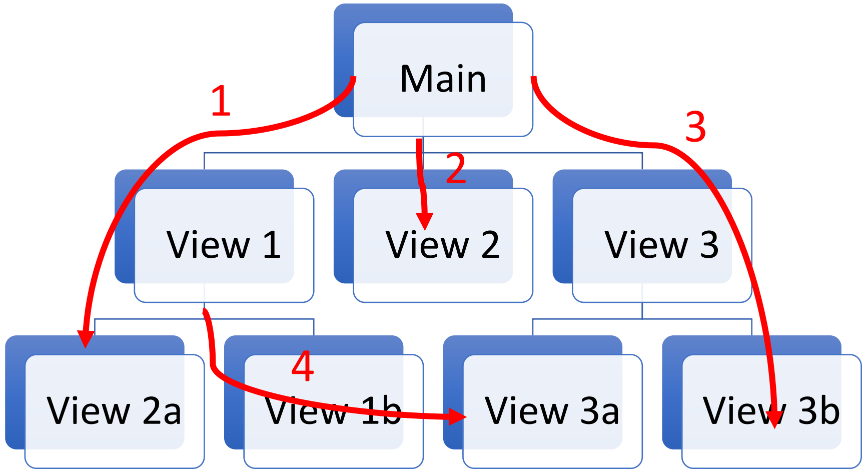
By Roane Dorado | ★★★★★ (1)

Use Creately's easy online diagram editor to edit this diagram, collaborate with others and export results to multiple image formats.

Storyboards

Storyboards

- Identify and chart out storyboards (***use-cases***) for all user subtasks or goals.
  - Note: it is possible to move between child views without returning to the root.
  - Add more functional detail to each of the blocks in the hierarchy
- You can use
  - A flow chart or,
  - Annotate the sitemap.
- An accompanying description of each use-case should be provided.
- Storyboards provide you with a strong grasp of user actions.



Motivation for Storyboards

- Why do this?
  - Identify common use-cases to make UX easier by creating shortcuts.
  - Identify unexpected use-cases, which can affect functionality.
  - Identify missed use-cases.

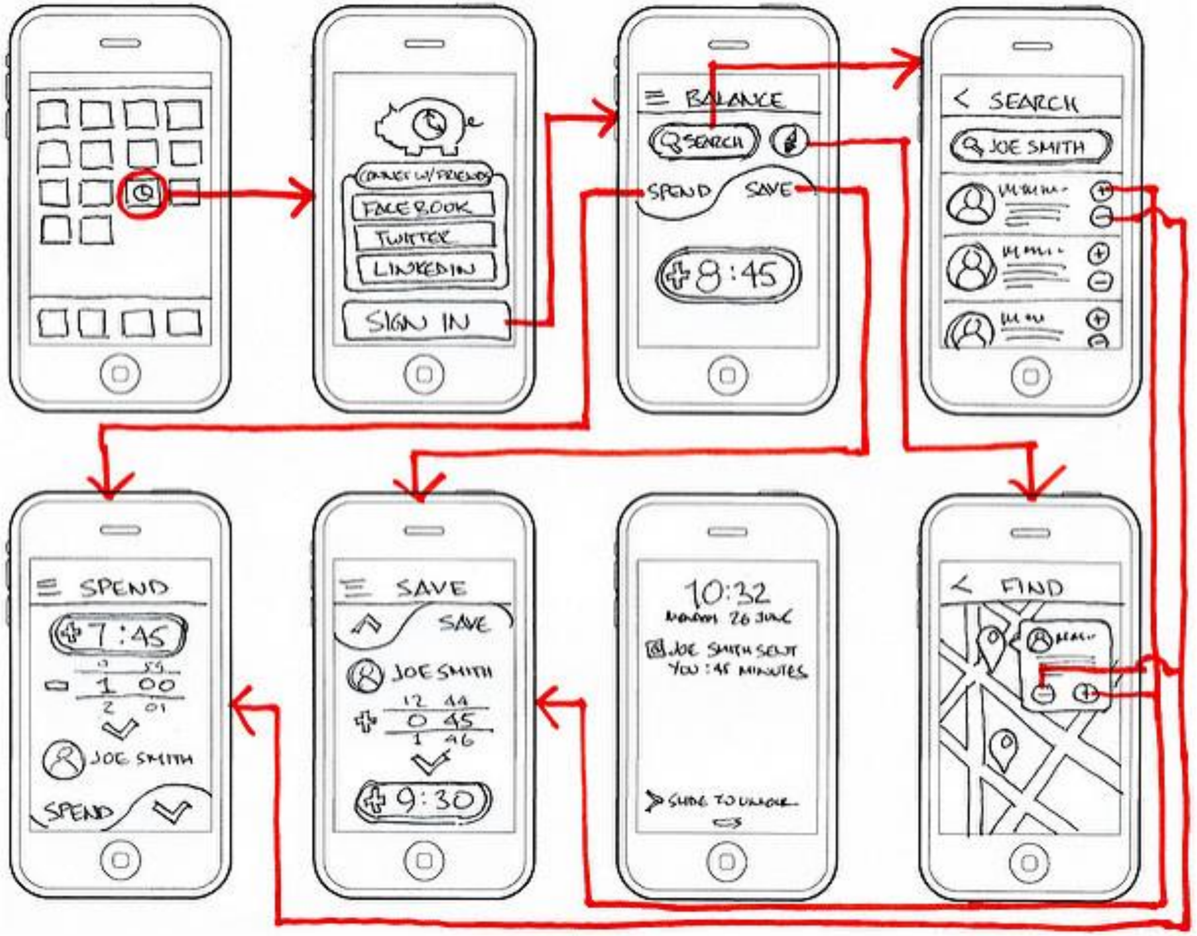
- Used to be called a “clickstream.” It describes the “use-case” or typical usage scenarios. It does not describe a feature but rather a process.

Sitemaps vs. Storyboards

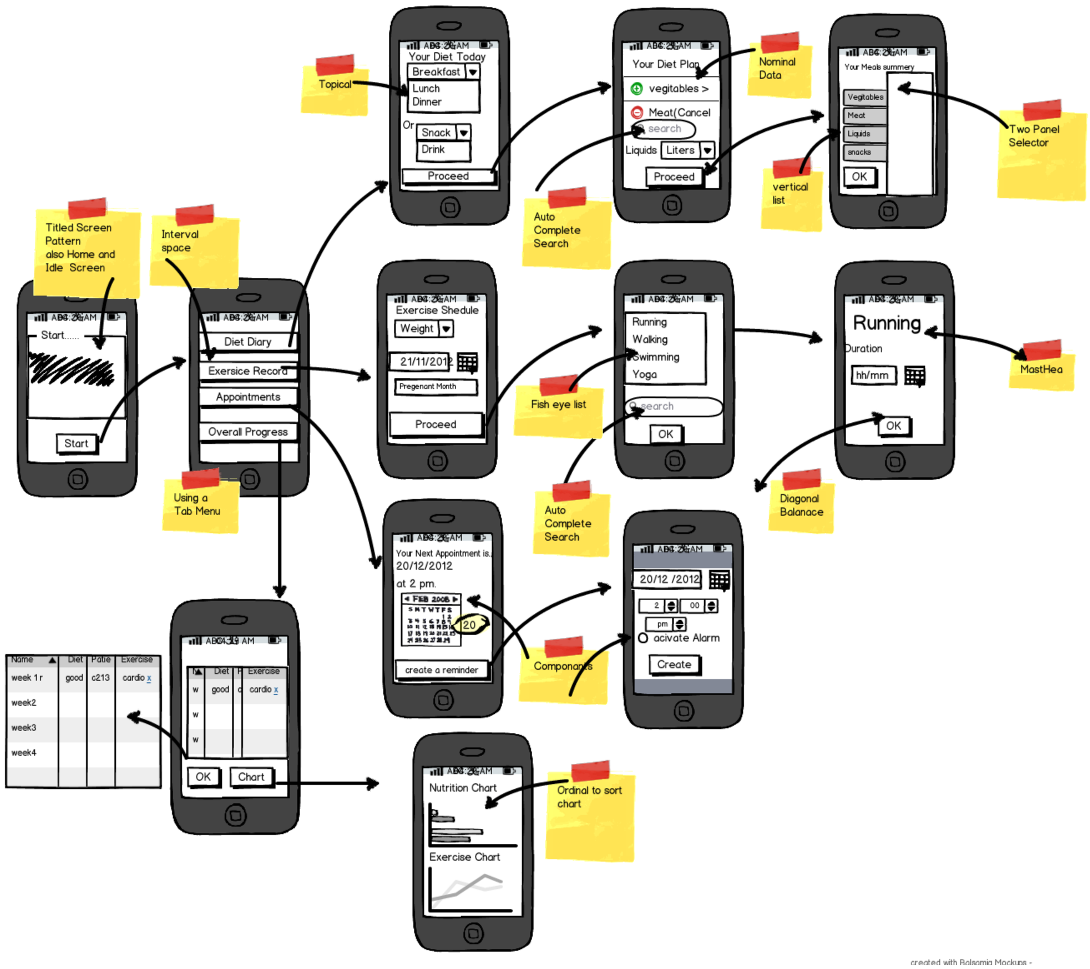
- The views, functionality, parts, and pieces of the application.
- The initial design of the UI.
- A static view of what the system has.
- Views are the foundation.

- The flow of usage between the pieces.
- The initial design of the UX.
- A dynamic view of how the system is used.
- Events and activities are the foundation.

- Developing these separately isn’t necessary. If we use an augmented sitemap, the two can be created at the same time.



Storyboard Example 1



Storyboard Example 2

Know your USERS (Target Audience)

Users probably expect to land on a “Splash Screen” or landing page.

- Do they expect to navigate from there immediately? (Probably if they are familiar with the site and have a task to do).
- Will they tolerate a fancy welcome graphic? (Maybe, depends on length, if they’ve seen it before, its appeal to them …)
- Will they log in? Only if there is some benefit they perceive.
- “Login with Facebook” – you tie yourself to a platform and risk being associated with them, might be easy but will also drive some number of users away – How many? More than you want?

Don’t just copy everyone else;

Do your (market) research and know your users.

Reverse Engineering your Storyboards

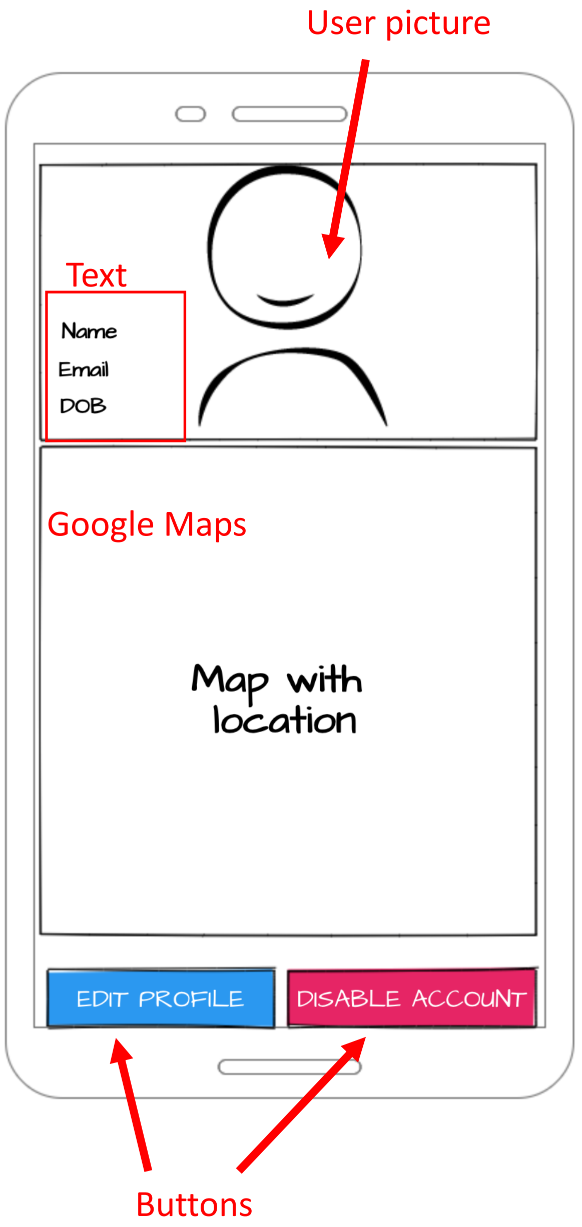
- Storyboards identify your use-cases/storyboards at the design phase.
- To test the design, we record clickstream data from user trials.
- Important to see: Are users doing what we intended or are they using the software differently (e.g., our understanding of them was wrong).
- Clickstream analysis is a new and growing methodology for understanding how an application is being used.
- BUT – Don't match the application to the user behaviour! Just because a user does something, it doesn't mean the user wants to do it that way!

Wireframes

Wireframes

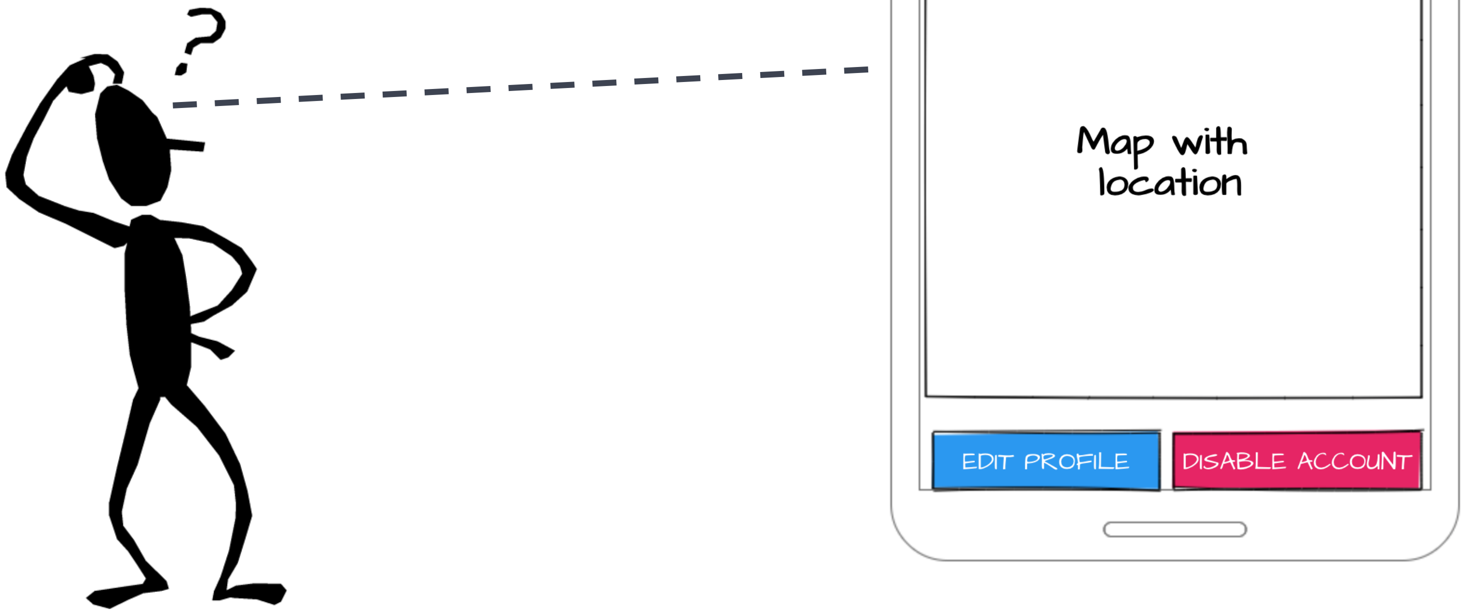
- A wireframe is a ***UI independent rendition of your content layout***.
  - “Back of the envelope” or “cocktail napkin” level of detail
  - Block drawing of each screen
  - Really just a low-fidelity form of prototype
  - Expect to change them often.

- Used to answer questions like:
  - How do we layout the navigation elements?
  - What visual / interaction metaphors should we use?
  - What are the best ways to present information?



Wireframes

Helps you think about how users will interact with your application, and plan so you can make interactions useful

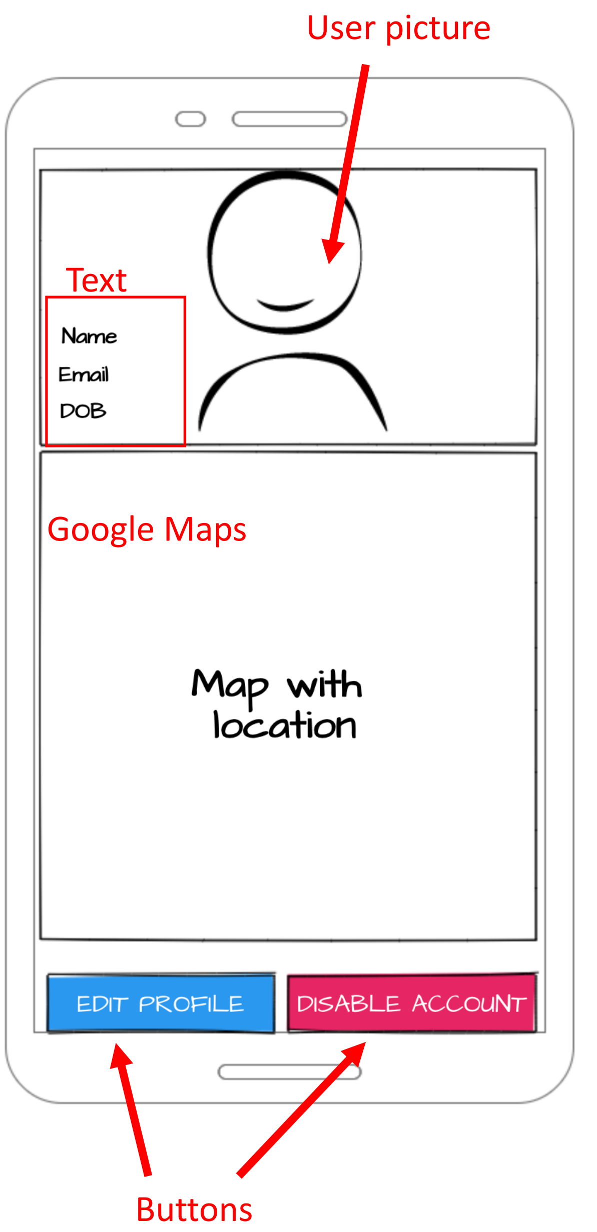


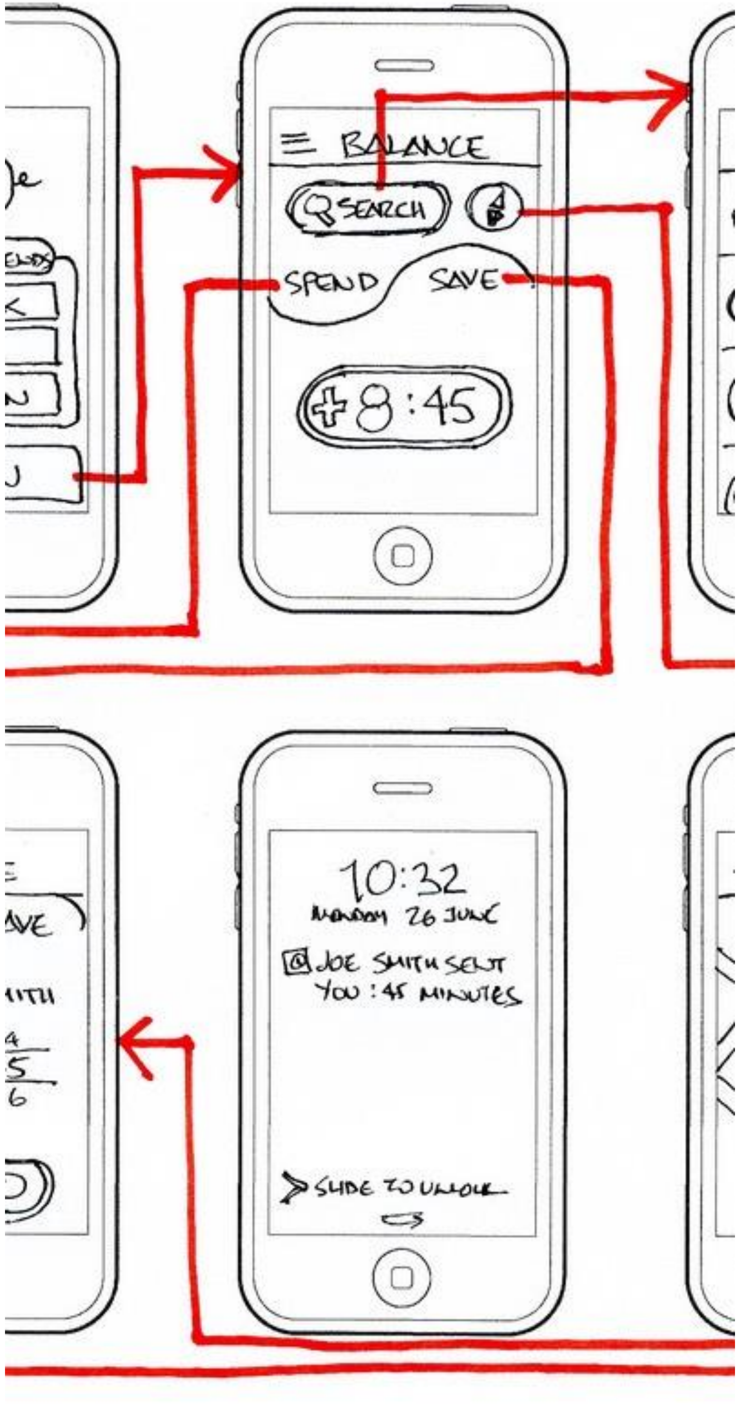
Wireframes

Notes:

- Don’t worry about UI details, e.g., colour, exact sizing, fonts
- Use annotations to describe interactions (it’s a communication tool).
- Use this stage to get feedback from others (e.g., from your users or clients)
- *Iterate through several layouts to find the right one*

Storyboards model function while Wireframes model presentation.





Wireframes

- Prioritize the information you will show on a page/view of the application
- Organise related content into blocks or chunks (grouping) – helps in “guiding” users to specific elements
- Paper + pencil → best / least expensive way
- You could also use various tools (online and offline) – some of them are free and some of them cost money
- You could also use PowerPoint, Photoshop, Illustrator, Paint, and other desktop tools to create wireframes
- Online tools: MockFlow and Draw.io

Prototypes

Prototypes

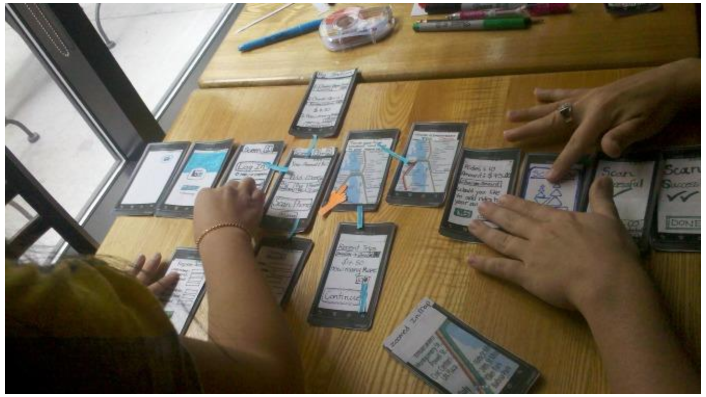
- ***A prototype is a model of what the end result should be like***.

- Prototypes come in many different forms and as the design advances more complex prototypes may be needed.

- Prototypes are used to:
  - Get users to try out your design.
  - Ensure that functionality and presentation come together properly.
  - Motivate a finer design of the app.
  - Provide a cheap way to iterate the design.

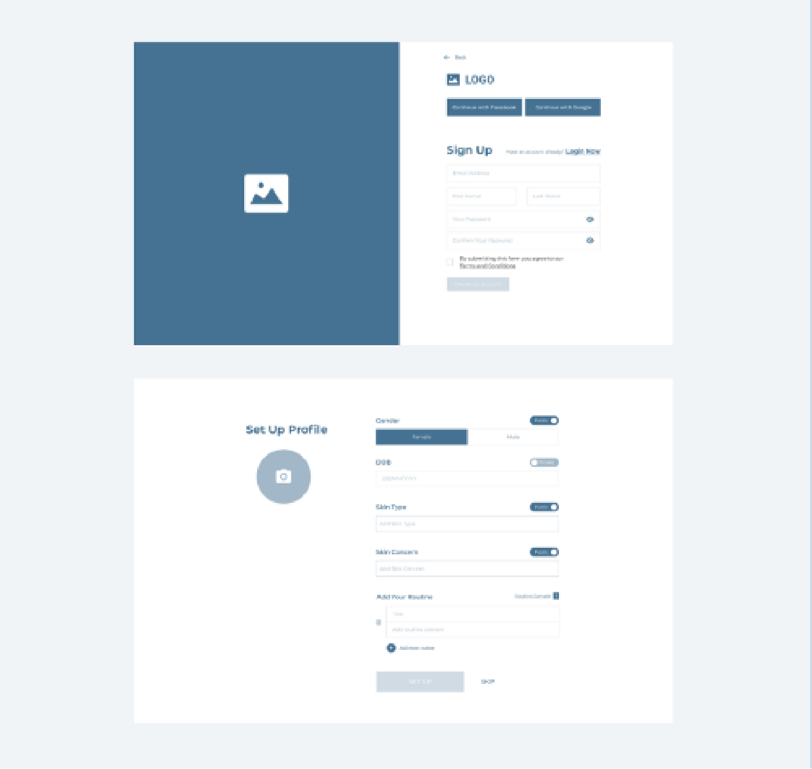
- First start with cheap low fidelity prototypes.

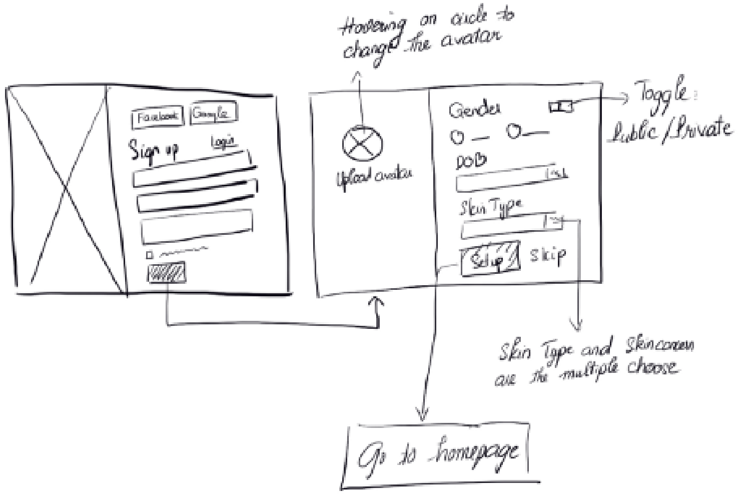
Low Fidelity Prototypes



- Do not model all details.
- Not very interactive.
- Are very cheap to produce and change.
- Sufficient to give any user an idea about how your app will work.
- Example 1: Take your wireframes and build on them; put them in a book.
- Example 2: ***Paper Prototypes***: scale size drawn screens.
- Example 3: PowerPoint drawn prototypes with slide links to move between screens.

Prototypes vs. Wireframes





Wireframes: Save time when making decisions. Help you communicate with the users before putting in more effort.

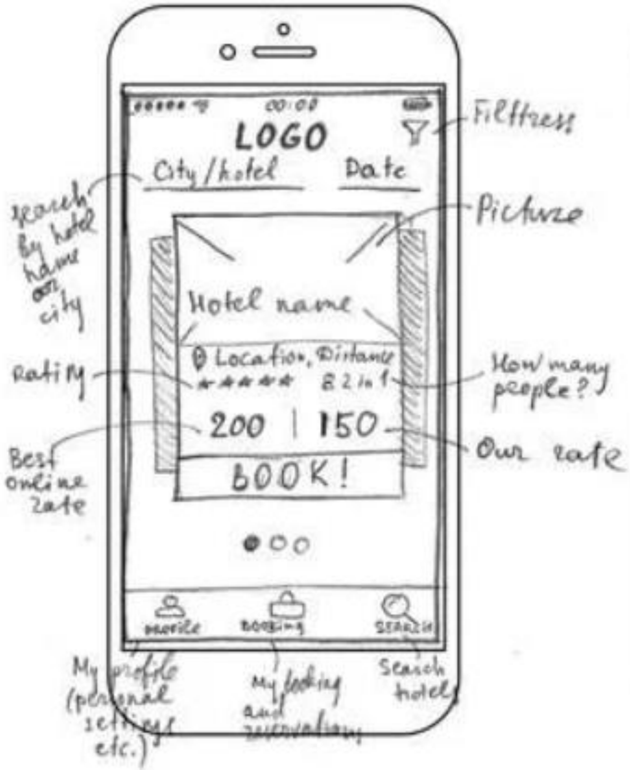
Disposable v. Evolutionary

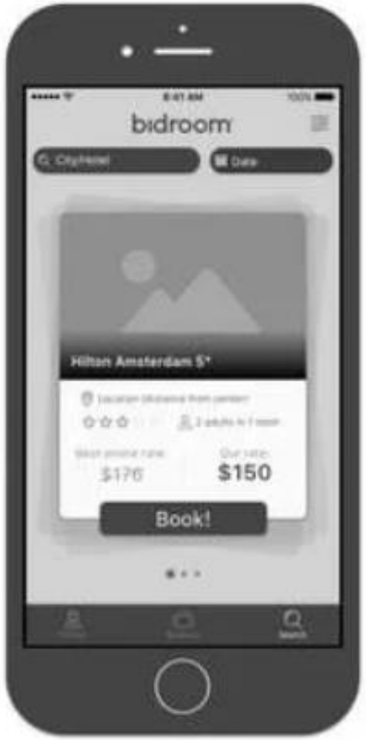
- Some prototypes are made to be thrown away.
- Example: making 3 different versions and selecting one of them; two of the versions are going to be discarded.
- Disposable prototypes let you explore, play, test, and not need to commit to anything at this point.
- Some prototypes are meant to evolve and become part of the end product.
- Code-based prototypes are often in this category.
- Once you get it right, you can start building on it.
- Sometimes forces you into directions you don’t want to go when a user sees them and thinks the prototype is a “finished product.”

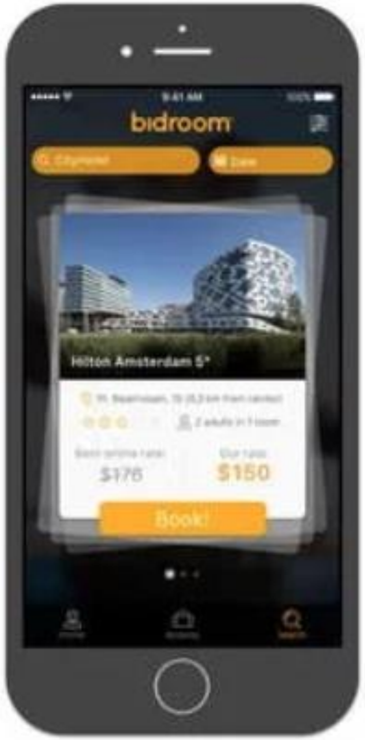
High-Fidelity Prototypes

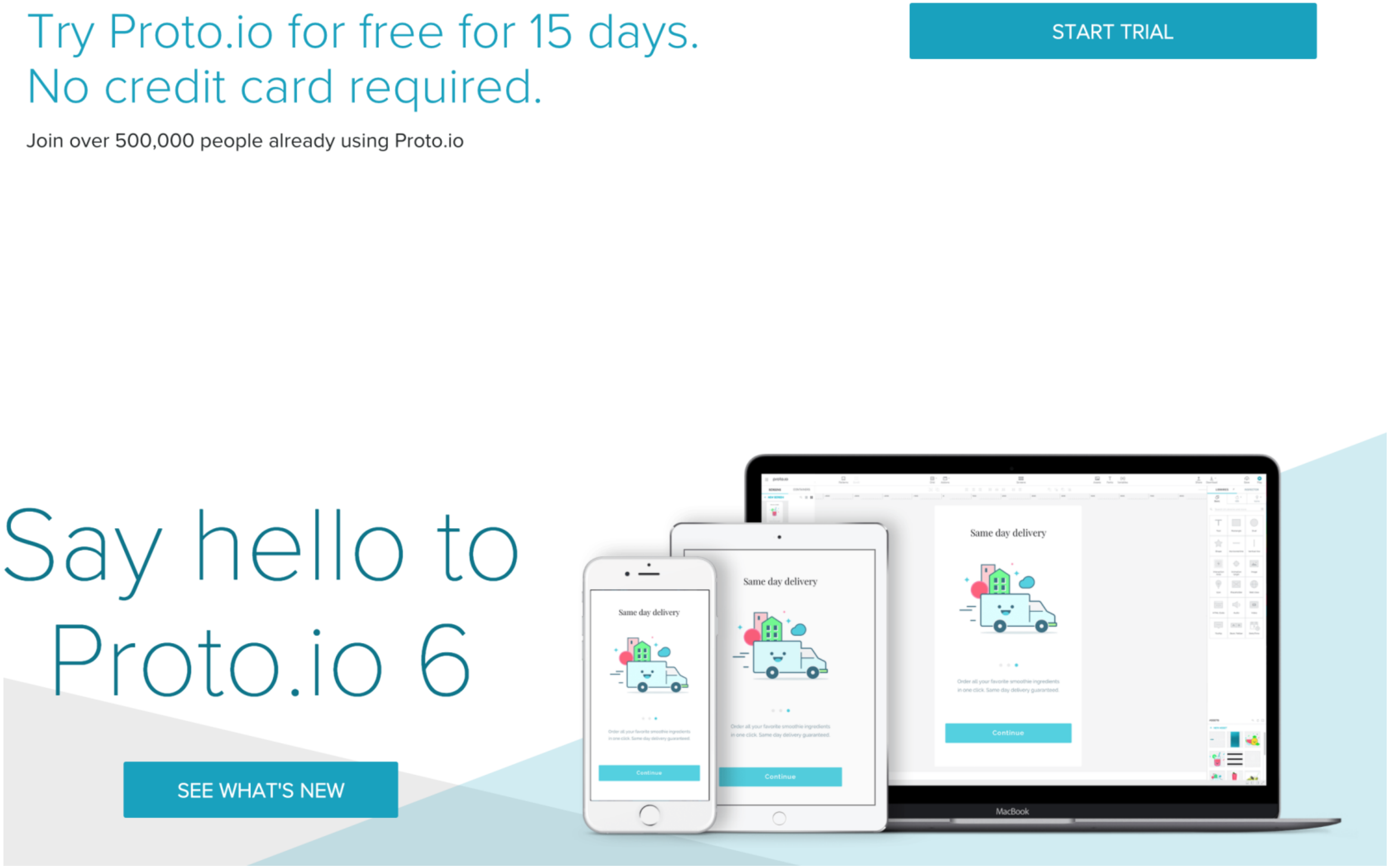
- Computer-based prototypes.
- Allow for a realistic (mouse/keyboard/touchscreen) user interactions.
- High-fidelity prototypes take you as close as possible to a true representation of the user interface.
- High-fidelity prototypes are assumed to be much more effective in collecting true human performance data (e.g., time to complete a task), and in demonstrating actual products to clients, management, and others.
- Functionality is often minimal or faked.
- You are still not committed to the design yet and want to maintain maximum flexibility to make changes.

Wireframe → Low-Fidelity Prototype (mock-up) → High-Fidelity Prototype









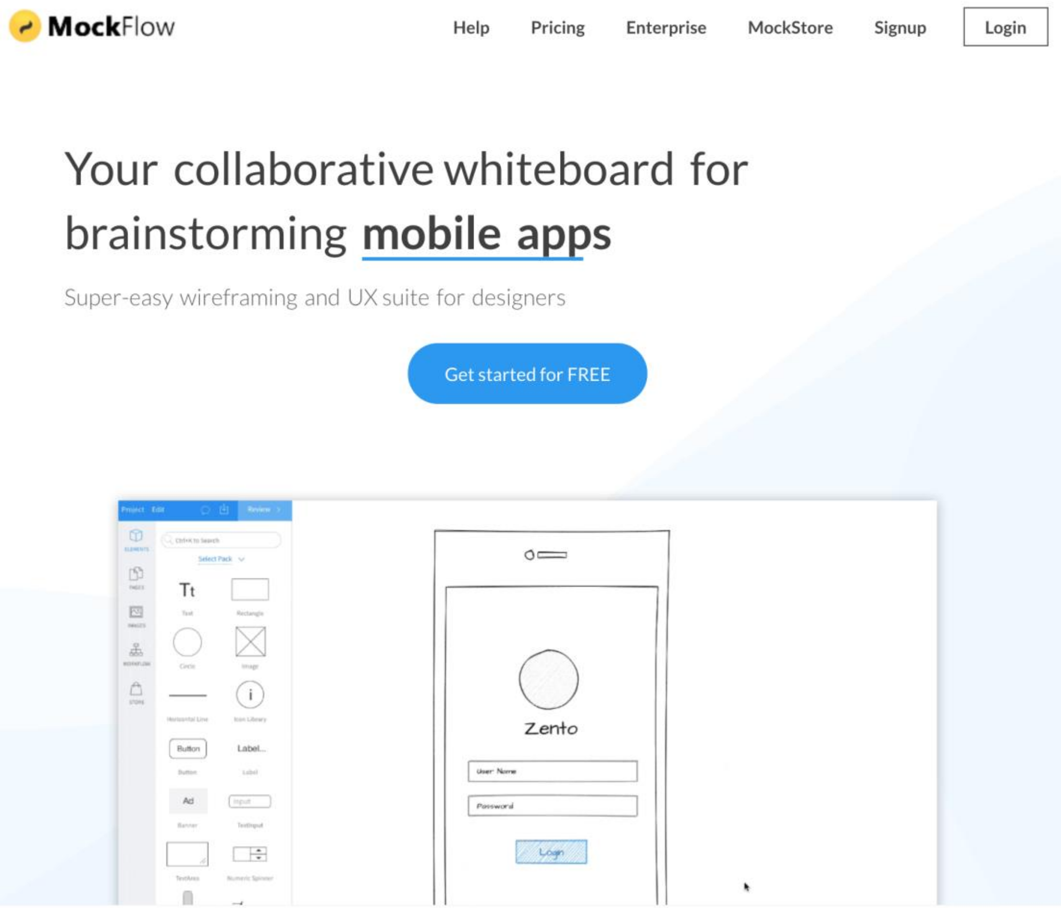
Try Proto.io for free for 15 days.
No credit card required.

Join over 500,000 people already using Proto.io

Say hello to Proto.io 6

Prototyping tools – proto.io

Prototyping tools – MockFlow



MockFlow

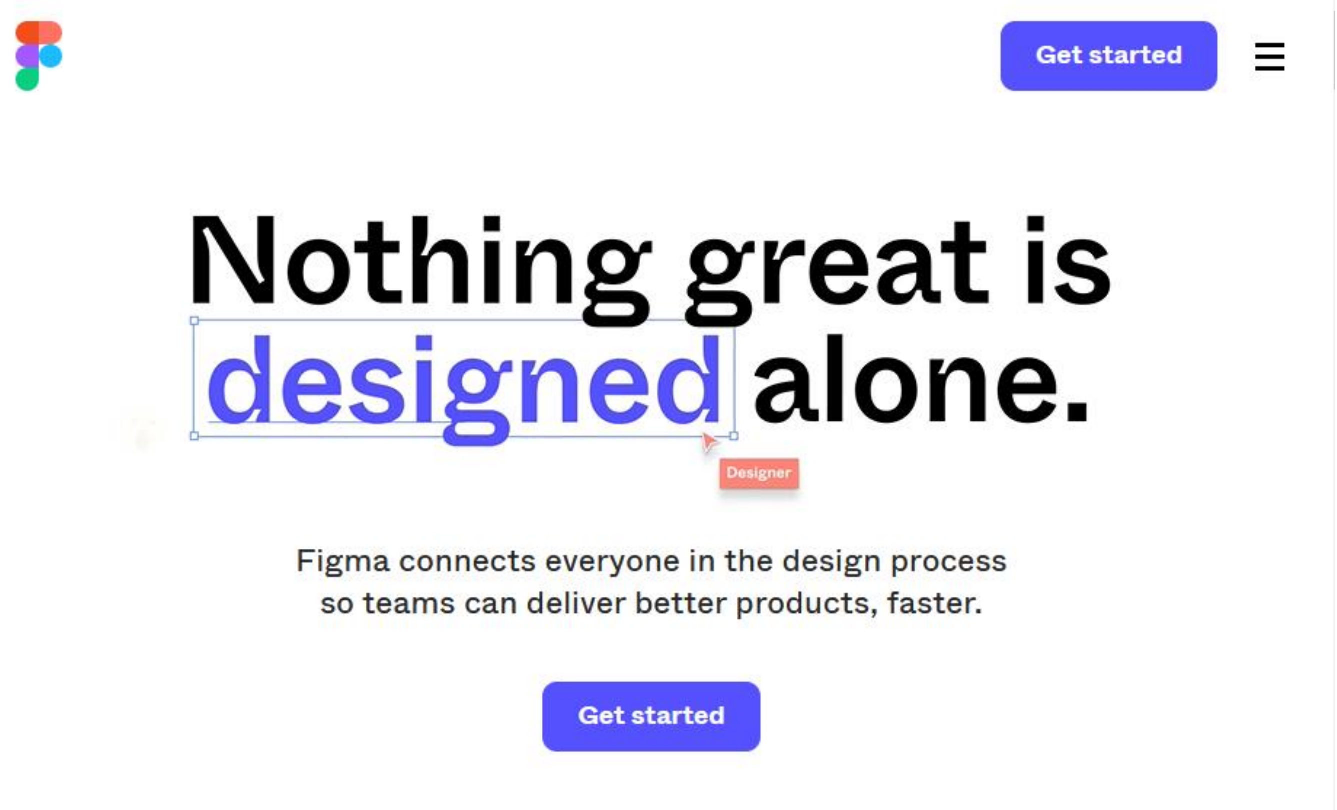
Help Pricing Enterprise MockStore Signup Login

Your collaborative whiteboard for brainstorming **mobile apps**

Super-easy wireframing and UX suite for designers

Get started for FREE

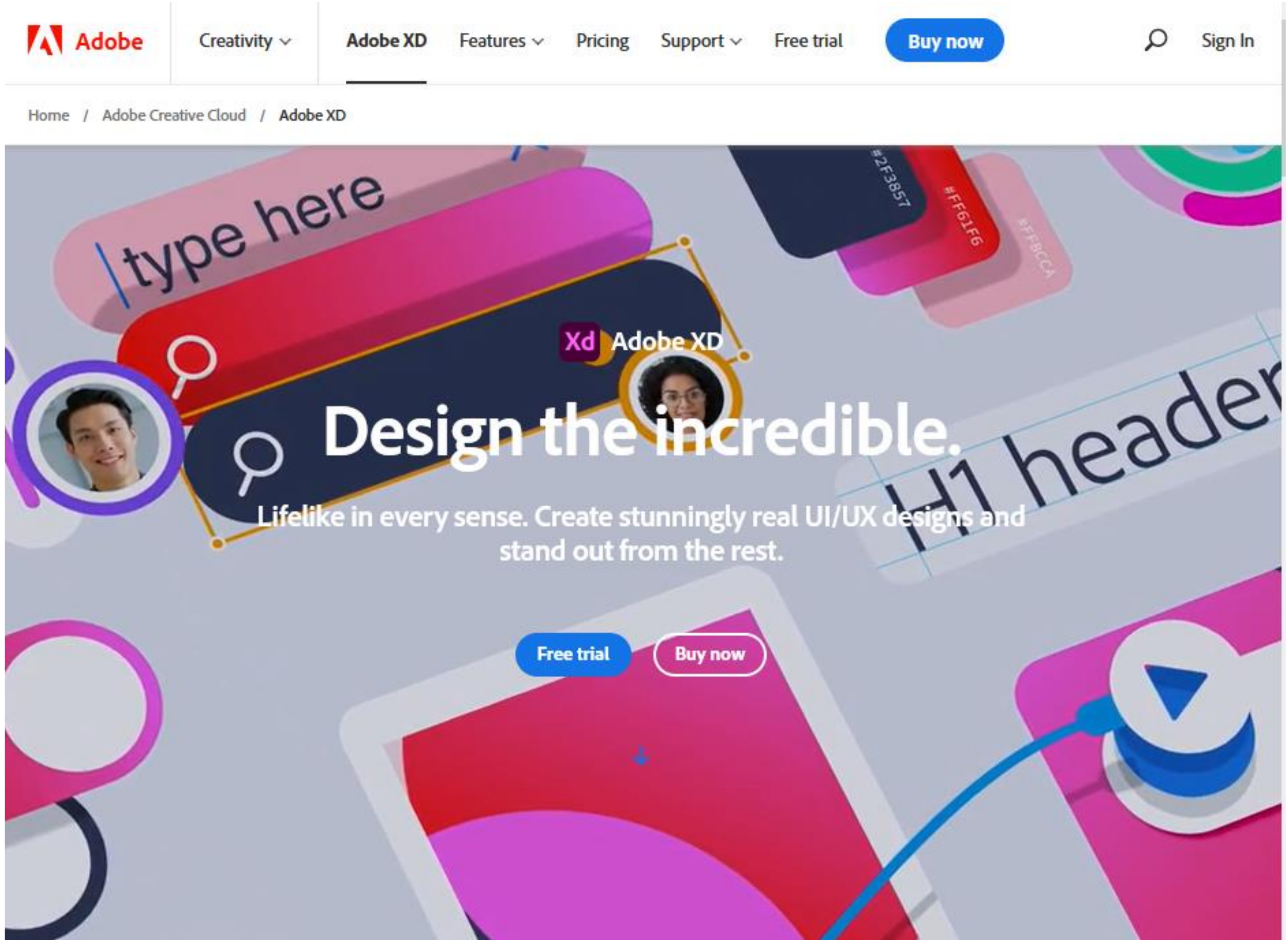
Prototyping tools – Figma



Nothing great is designed alone.

Figma connects everyone in the design process so teams can deliver better products, faster.

Prototyping tools – Adobe Xd



Adobe Creativity Adobe XD Features Pricing Support Free trial Buy now Sign In

Home / Adobe Creative Cloud / Adobe XD

Adobe XD

Design the incredible.

Lifelike in every sense. Create stunningly real UI/UX designs and stand out from the rest.

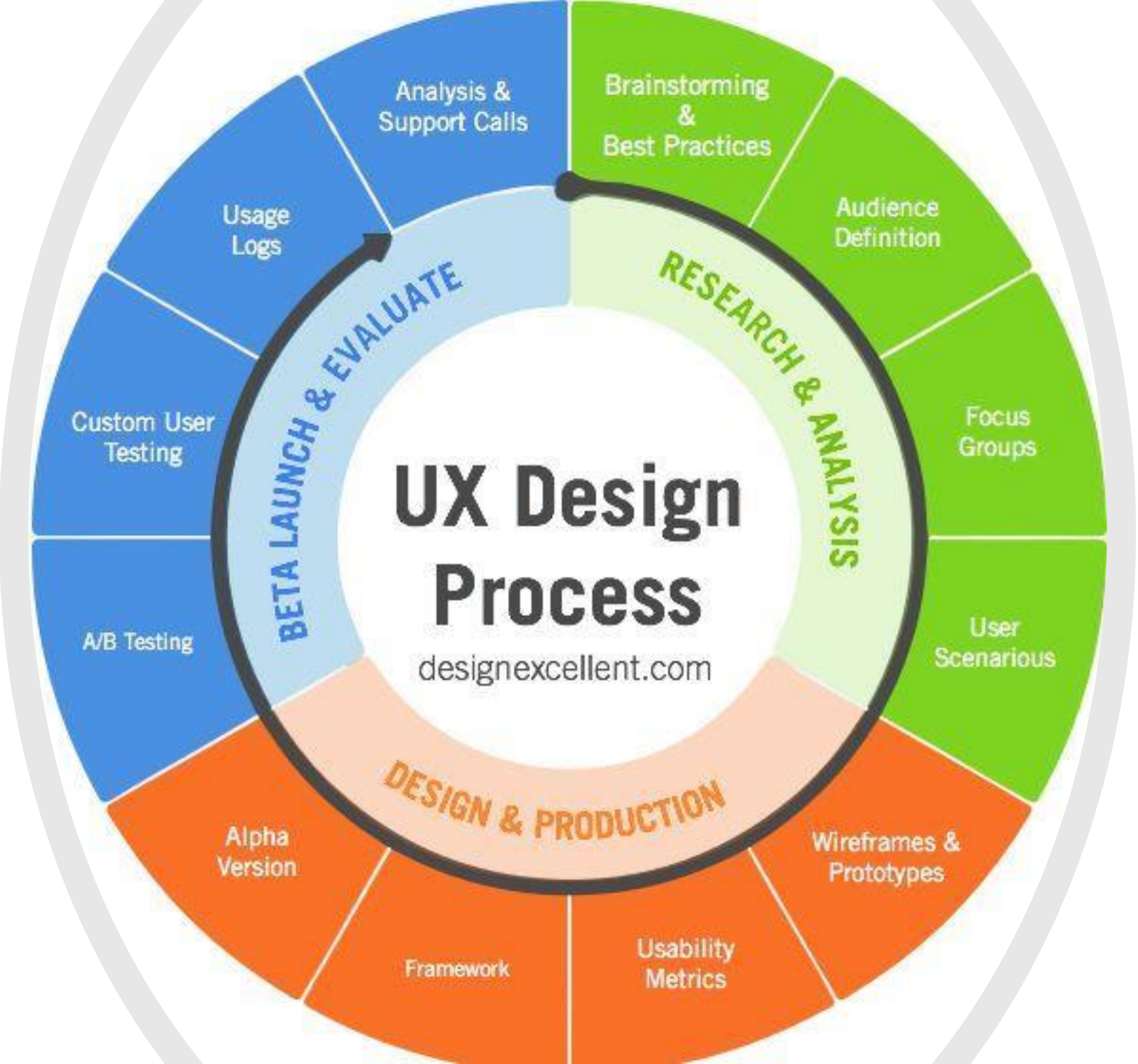
Free trial Buy now

Summary:
Simplified UI/UX Development Procedure

1. Use-Case: Know what your users want to do.
2. Sitemap: Determine how all the information is related.
3. Storyboard: Examine how use-cases work on the site map.
4. Wireframe: Decide roughly what each screen looks like.
5. Prototypes: Add detail and refine the wireframes.
6. Implement: Create a version for the application.

Evaluate & Test:

Frequently, at every step, with your target audience.



UX Design Process

designexcellent.com

RESEARCH & ANALYSIS

Brainstorming & Best Practices

Audience Definition

Focus Groups

User Scenarious

DESIGN & PRODUCTION

Wireframes & Prototypes

Usability Metrics

Framework

Alpha Version

BETA LAUNCH & EVALUATE

A/B Testing

Custom User Testing

Usage Logs

Analysis & Support Calls