HCI Lecture 4

Requirements and Prototyping

Todays’ plan

- Admin
  - The Gooey GUIs
  - Project stuff
- Lecture
  1. User surveys and data analysis
  2. Functional and Non-Functional Requirements
  3. Prototyping
  4. Usability Heuristics
  5. Design Principles

- Gooey GUIs and traffic light system
- Glossary/Weblinks
- Monday 25th April tutes

Project

- Worksheet 01 submissions – personas, user surveys, user stories
- Background – introduction, context, importance, what has already been done
- Assignment :
    - Explain and analyse your pain points, ‘how might we…’ problem statements, and how you iterated on your solutions
    - What you end up with looks nothing like where you begin
    - Take notes and record discussion
- Remember – should be using FigJam to do your brainstorming! Templates
- Screenshot FigJam into Assignment and explain your analysis
- Check the ‘*Women’s mentoring app*’ case study

Workshop 02 – just completed!

1. **Pain points – patterns**? Organise into groups (affinity mapping) - converging
2. Define your **problem statement: How might we…..<>**

   https://www.nngroup.com/articles/how-might-we-questions/

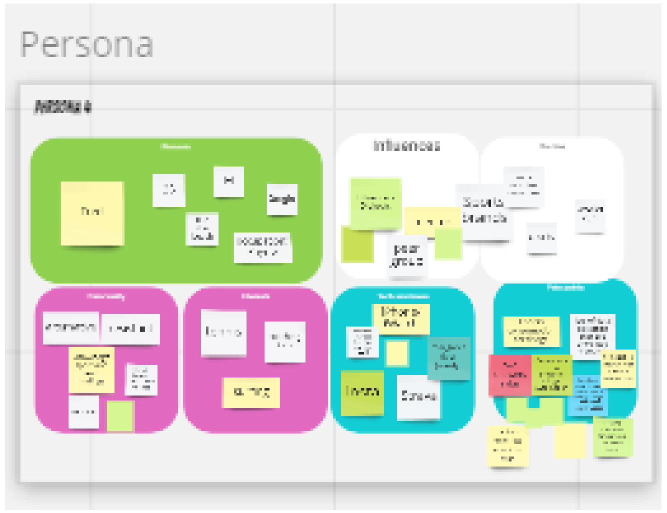
3. **Ideate solutions and group themes** (affinity mapping)

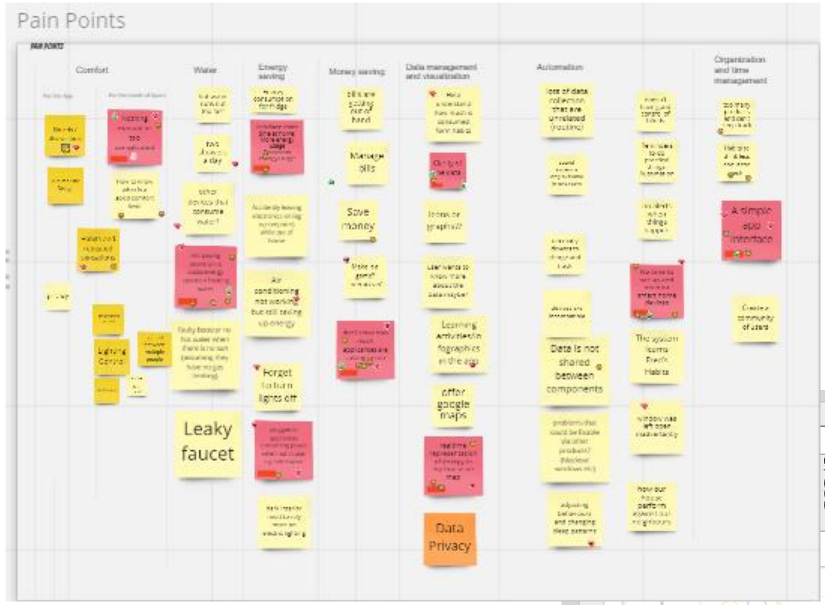
- **Software engineering methodologies**
  - Choose tools and frameworks (Agile, Kanban etc.)
  - Start employing them

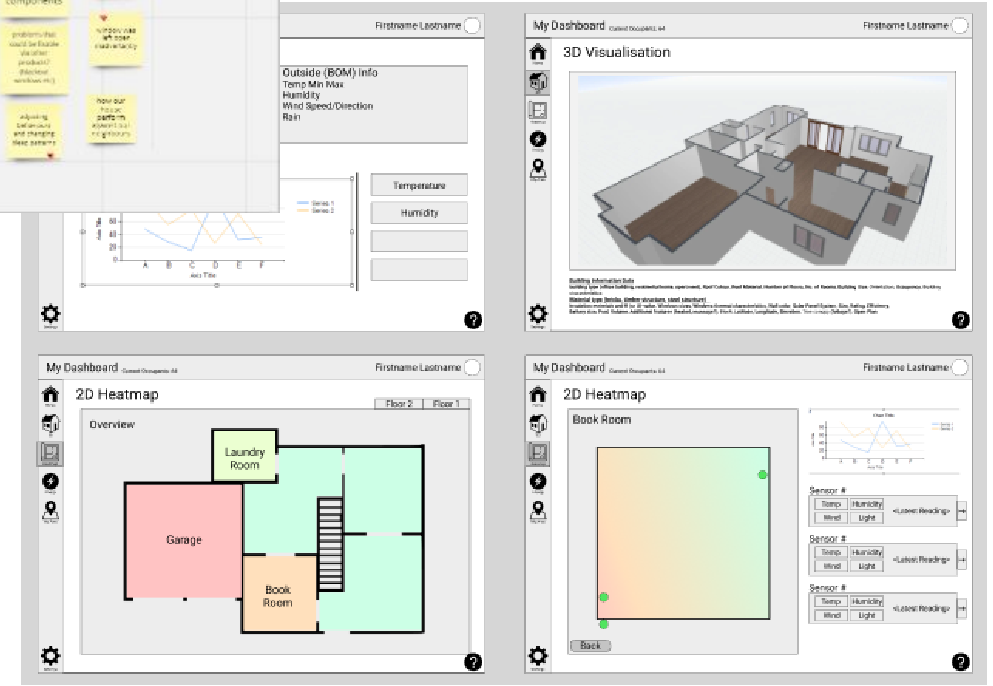
Recap: Use-case – digital twin project

- Miro:
  - Pain points
  - How might we…
  - Solutions
  - Best in world – Competitor analysis:
    - Strava, Uber

- FigJam:
  - Low-fidelity prototyping
  - High-fidelity prototyping




Pain Points
Comfort
Water
Energy saving
Money saving
Data management and visualisation
Automation
Organization and time management
Leaky faucet
Save money
Manage bills
Data Privacy
Forget to turn lights off
A simple app interface
Data is not shared between components



User Interviews and surveys

- Interviews/surveys
    - Structured/Unstructured/Semi-structured
    - Open questions
    - Closed questions easier to analyse
    - Don’t lead
    - Simple questions
    - Unconscious bias
    - Sensitive issues - anonymity
    - Prototype, prop, scenario
    - Test with a pilot group to make sure it is understood

Chapter 8: Data Gathering, Interaction Design (5th ed.) by Sharp, Rogers, Preece

Many other user-experience research methods

- User observation
- Group interviews
- Documentation
- Think aloud evaluation
  - https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/
- Evaluation of working prototype
- Diary/camera studies
- Usability tests
- Ethnographic field studies
- Eye-tracking

Chapter 8: Data Gathering, Interaction Design (5th ed.) by Sharp, Rogers, Preece; and https://www.nngroup.com/articles/which-ux-research-methods/

Qualitative and quantitative methods

- **Quantitative data** (How many? How much?)
  - Indirect methods
  - Measurement and mathematical analysis
  - Using a tool such as a survey

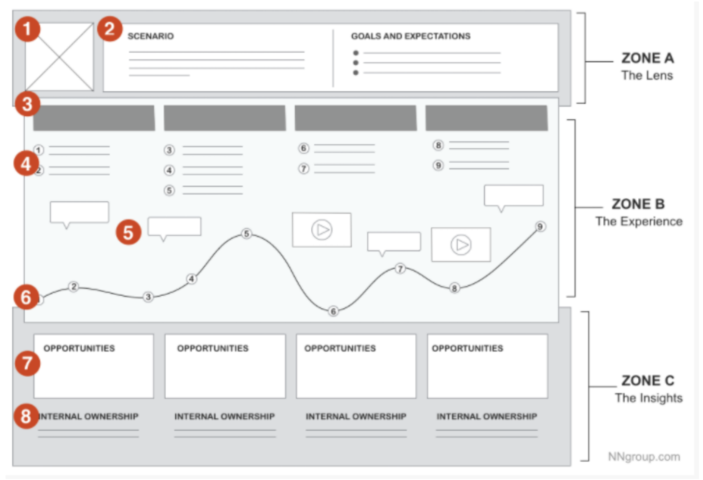
- **Qualitative** (Why? How?)
  - Direct method
  - Data about behaviour and attitudes via observation
  - Nature of elements, themes, patterns, stories, key events.

https://www.nngroup.com/articles/which-ux-research-methods/

Requirements

Refer to your notes from: “Intro to SE” ISAD 1000

- Explore the problem space
- Establish a description of what will be developed
- **Define requirements using:**
  - **Persona:**
    - Small set of personas with one primary
  - **Scenario:**
    - When, where, and how the story of the persona takes place.
    - **Use case:** description of scenario, UX, action, system response
      - **Use case specification:** textual
      - **Use case diagram:** use cases, actors, flow of events
    - User journey: persona, emotion, needs



User journeys
https://www.nngroup.com/articles/customer-journey-mapping/

Functional and Non-Functional Requirements

Functional Requirements

What does the system *do?*

Non-Functional Requirements

What should the system *be?*

- Software Requirements Specification
- Agile methodology (Backlogs (evolve/added) or WBS (fixed))
  - Kanban, Trello, etc.

Functional requirements

Recipe Keeper 17+
Meal planner & shopping list
Tudorspan Limited
★★★★★ 4.8 • 6.9K Ratings
Free · Offers In-App Purchases

https://apps.apple.com/us/app/recipe-keeper/id974683711

- **Modular functions** that the software should *do*
- Consider a recipe organiser app
- User requirements – directly observable by user
  - *The user must be able to calculate their total calorie consumption at all times*
  - *The user must be able to sort and 'like' all suggested daily recipes from the home screen.*
- System requirements
  - *The system must be able to register a new person to join a family account*
  - *The system must be able to offer several breakfast, lunch, dinner recipes every day.*
  - *The system must be able to convert recipes into shopping lists for a major grocery store*

Basic example of what **you need to do for functional requirements**

(Yours will be more specific)

- **User story**: As <persona>, I want <behavior> so that <benefit>

“As Mary, I want to organise my recipes, so I can pick a variety of healthy, cost-efficient recipes”

- **User functional requirements**:
  - *The user must be able to sort through recipes using search categories such as meat/vegetarian/vegan, calories, cost, seasonal produce.*
- **System functional requirements**:
  - *The system should recommend a list of healthy daily recipes.*
  - *The system must provide a shopping list function.*
  - *The system must be able to suggest an alternative recipe if user does not have ingredients available.*

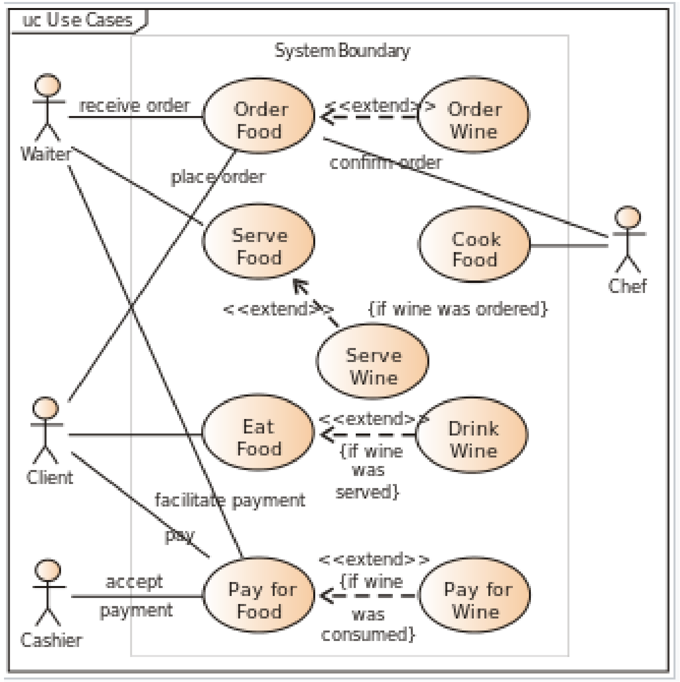
Some ideas here: https://www.nuclino.com/articles/functional-requirements

Functional requirements can also be represented by:

1. Use case diagram
2. Use case specifications (textual)

Use case (Happy Days version)

1. System asks for user login details
2. User provides login details
3. System asks user which day to calculate total calorie intake for.
4. User specifies day.
5. System returns total daily calorie intake.



https://en.wikipedia.org/wiki/Use_case_diagram

Non-Functional Requirements

- **Non-functional requirements** – what should the system *be?*
  - System features as a whole:
    - **Usability:** UI, data presentation, learnability, languages, dates
    - **Performance:** response time
    - **Reliability:** reduce failure
    - **Security:** data privacy and loss
    - **Platform:** compatibility
    - **Software:** OS, browser
    - **Scalability, interoperablility,** etc …..

  - Can be represented by user stories
  - Can be attached to use cases

Non-Functional requirements

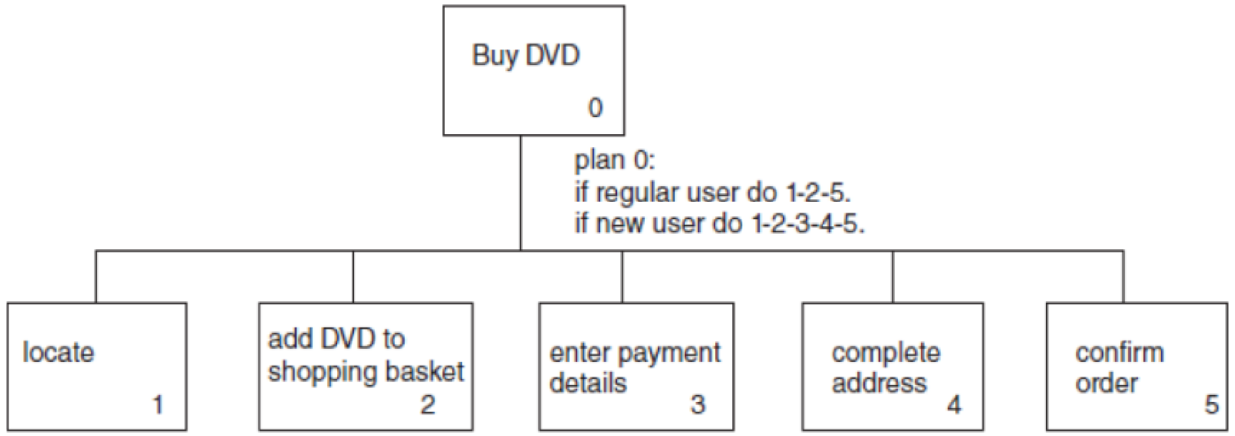
- “The system must….”
- “The user must... ”
- Examples:
  - *The system must allow the user to be aware of activity status (browsing, check-out etc.) at all times.*
  - *The system must allow the user to reach the payment screen in three mouse clicks.*
  - *The system must allow the user to understand options without being able to read English.*
  - *The system must return recipe suggestions to the user in <1s.*
  - *The system must have a recipe retrieval failure rate less than 1%*
  - *The user should not re-use passwords.*
- Can also be written as user stories
- Not modular tasks as in the way functional requirements are
- Must be specific
- System wide attributes
- Qualitative but objective (testable)

Worksheet 03/Assignment 1 - Requirements

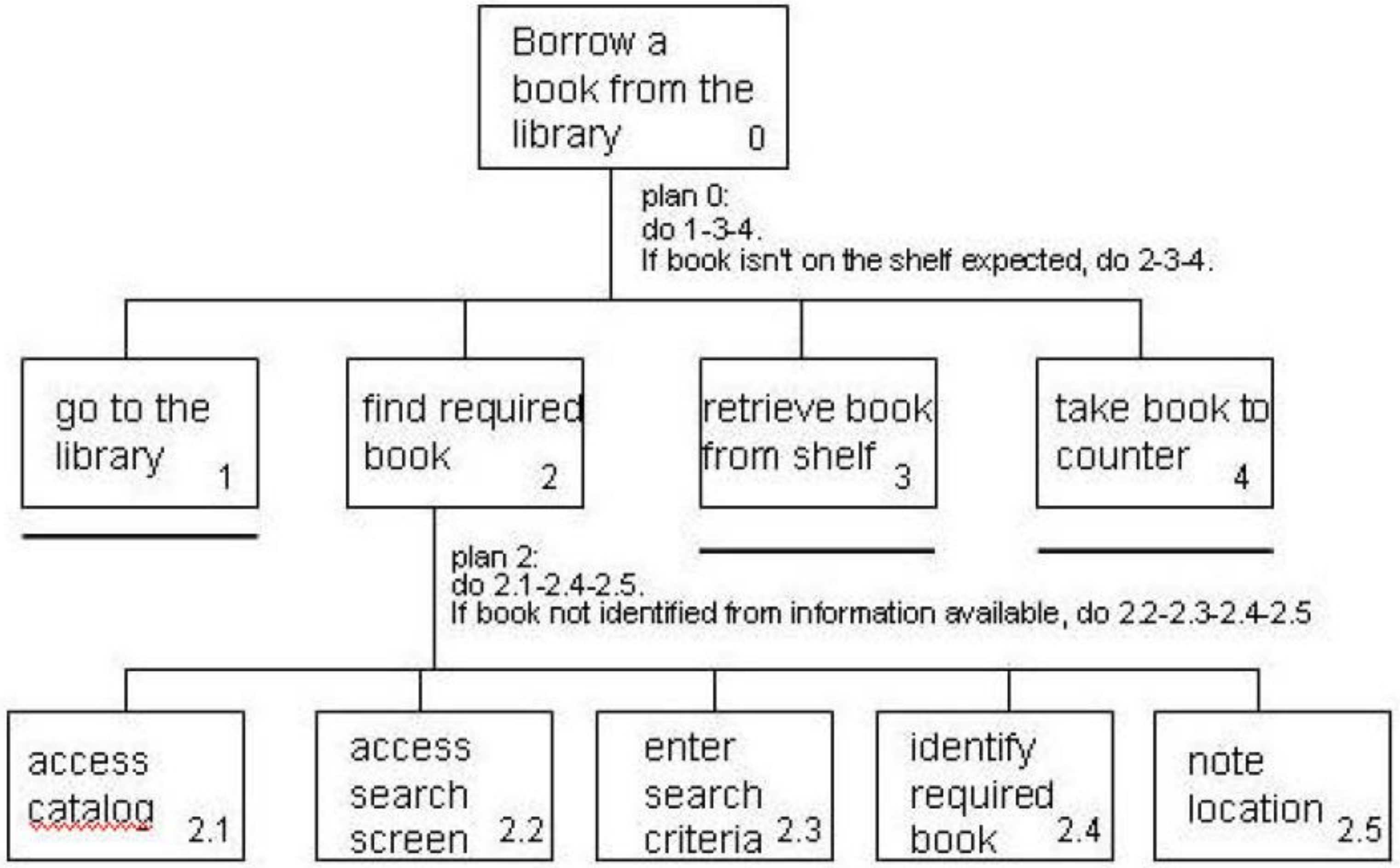
- Use your judgement
- Don’t have to list every single requirement – some are trivial and understood
- Ensure these requirements are good quality, well-thought out and specific to the key functionality you are designing in your app.
- Think about your main NFRs and importantly what usability NFRs you would like your system to have

Hierarchical task analysis (HTA)

- Identify several **key user activities** that your user will perform in your app.
- Perform HTA
- Breaking tasks down into subtasks, then sub-sub-tasks, then sub-sub-sub tasks which are grouped as plans
- Also check out: https://www.nngroup.com/articles/task-analysis/


Buy DVD
0
plan 0:
if regular user do 1-2-5.
if new user do 1-2-3-4-5.
locate
1
add DVD to shopping basket
2
enter payment details
3
complete address
4
confirm order
5

Hierarchical Task Analysis



Borrow a book from the library 0
plan 0:
do 1-3-4.
If book isn't on the shelf expected, do 2-3-4.
go to the library 1
find required book 2
retrieve book from shelf 3
take book to counter 4
plan 2:
do 2.1-2.4-2.5.
If book not identified from information available, do 2.2-2.3-2.4-2.5
access catalog 2.1
access search screen 2.2
enter search criteria 2.3
identify required book 2.4
note location 2.5

Why Prototype?

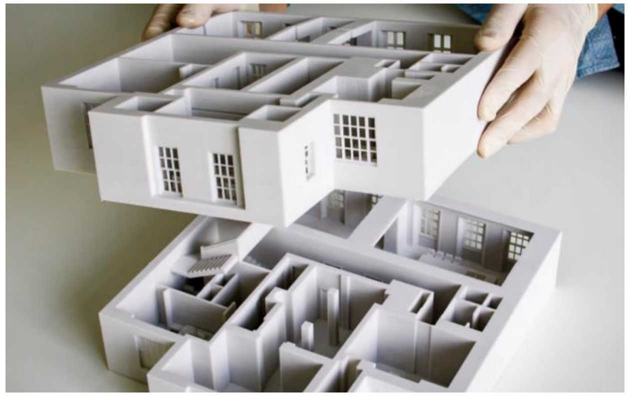
- Evaluation and feedback
- Stakeholders can interact with prototype
- Team members communicate
- Ideas tested
- Reflection
- Answer questions and support designers in choosing between alternatives

Prototyping

- Small-scale, low cost, flexible
  - 3D printing
  - car/architecture models
  - Alternative designs – change out features
  - 3D visualisation

https://www.rapidmade.com/rapid-prototyping

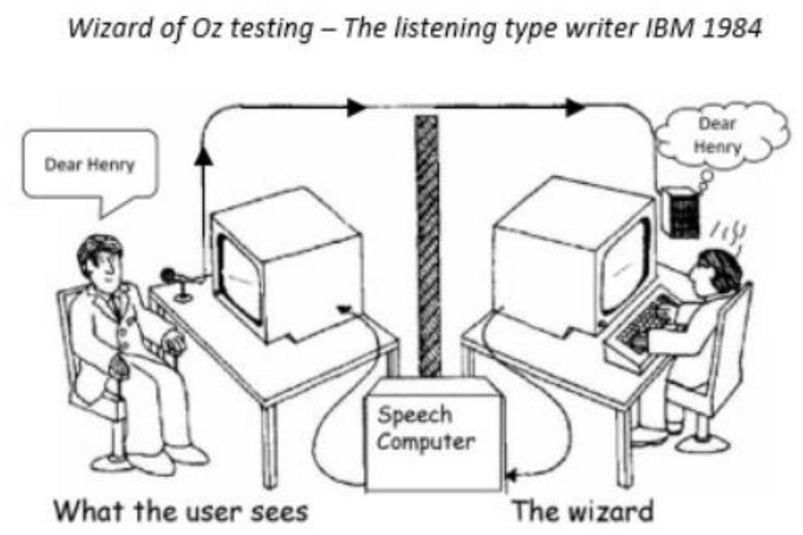
https://group.mercedes-benz.com/innovation/design/design-process.html



https://all3dp.com/1/how-to-make-a-3d-printed-architecture-model/

https://www.auganix.org/kia-motors-driving-the-future-of-car-design-with-varjos-collaborative-xr-technology/

Prototyping



Wizard of Oz testing – The listening typewriter IBM 1984

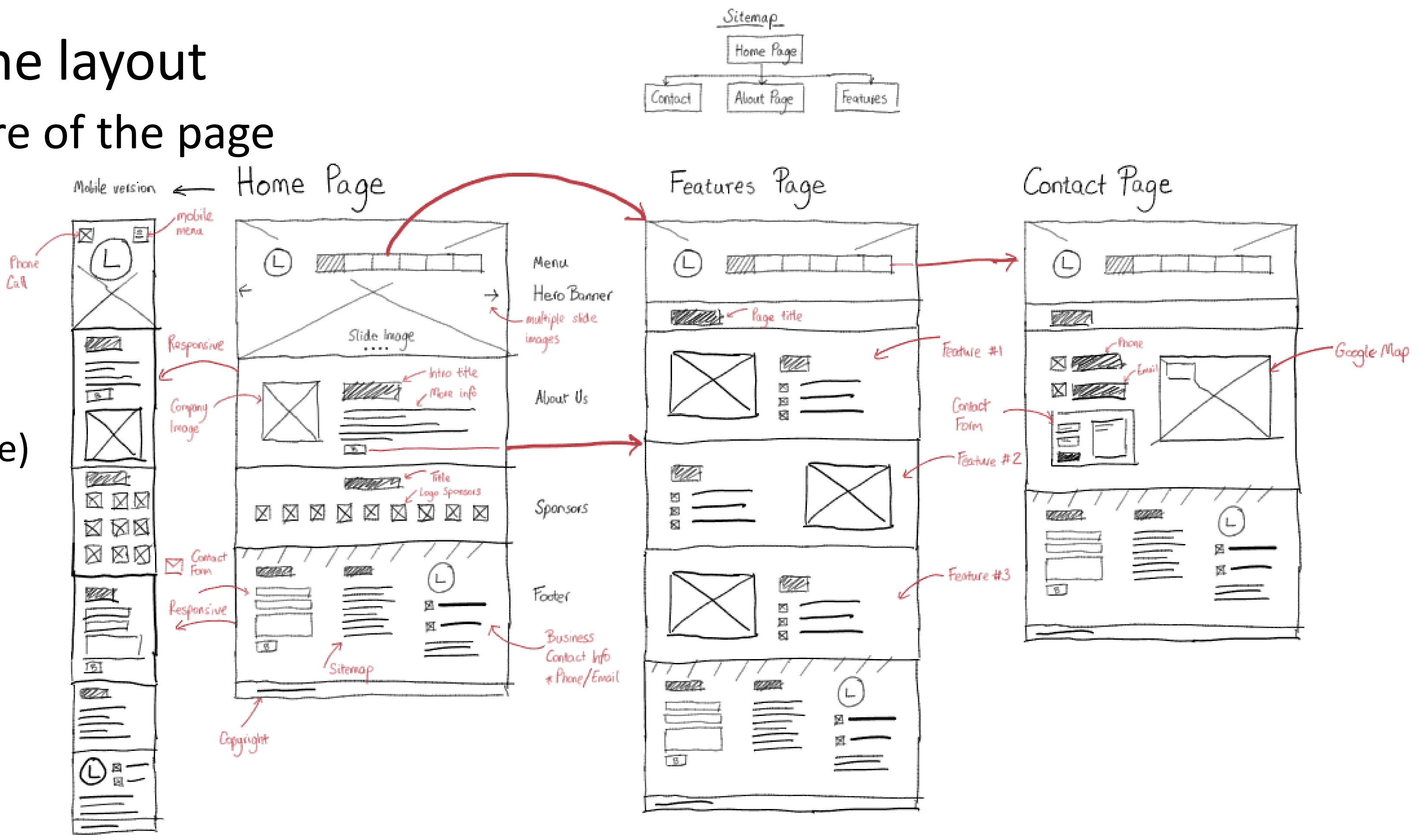
- Sketches of screens, task and window sequences
- Paper-based – cheap and easy
  - Sketching
  - A4, post-it notes, flash cards
- Story-boarding (~user journey)
  - https://www.nngroup.com/articles/storyboards-visualize-ideas/

- Mood board
  - https://xd.adobe.com/ideas/process/ui-design/how-to-enhance-ux-design-with-mood-boards/
  - https://www.canva.com/learn/make-a-mood-board/

- Wizard of Oz
  - *Wizard fools everyone by creating a vision of himself to look powerful by using a set of controls while he hides behind a curtain concealing the reality*
  - https://ux4sight.com/blog/wizard-of-oz-prototyping

Wireframing

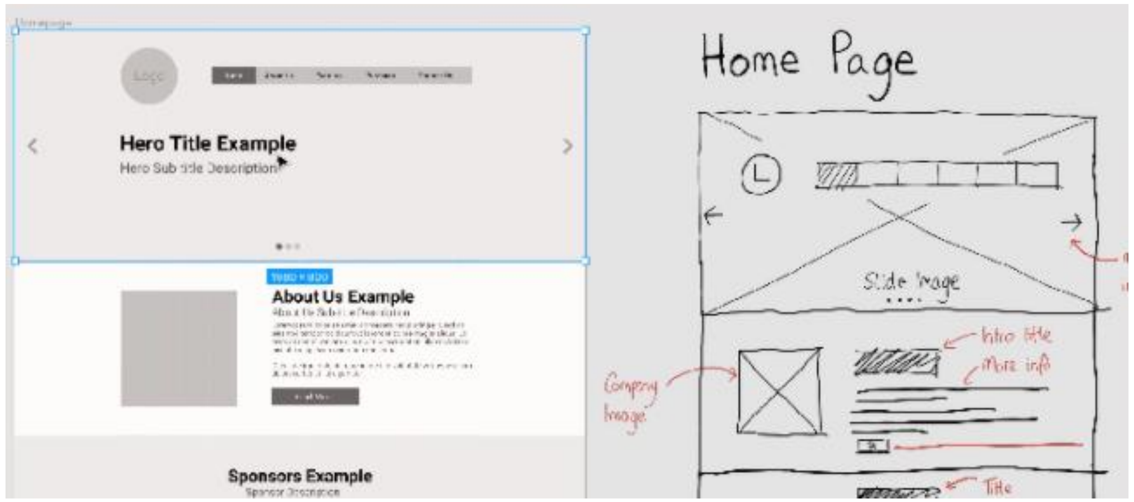
- Wireframing – sketching the layout
  - Understand overall structure of the page
  - Navigation
  - Paper and pen sketch
  - Tablet or free online tool
    - Figma,
    - MockPlus, Balsamiq (not free)
  - Iterate over and over….



https://www.freecodecamp.org/news/ui-ux-design-tutorial-from-zero-to-hero-with-wireframe-prototype-figma/
https://careerfoundry.com/en/blog/ux-design/free-wireframing-tools/

Low-fidelity UI/UX Prototyping

- Use Figma
- Built on top of wireframe
- Include text
- Limit colour and images
- Pay attention to font size, spacing, content location, margins and padding
- Layer groups and sections (Header section, About Us section)
- Iterate over and over



https://www.freecodecamp.org/news/ui-ux-design-tutorial-from-zero-to-hero-with-wireframe-prototype-figma/

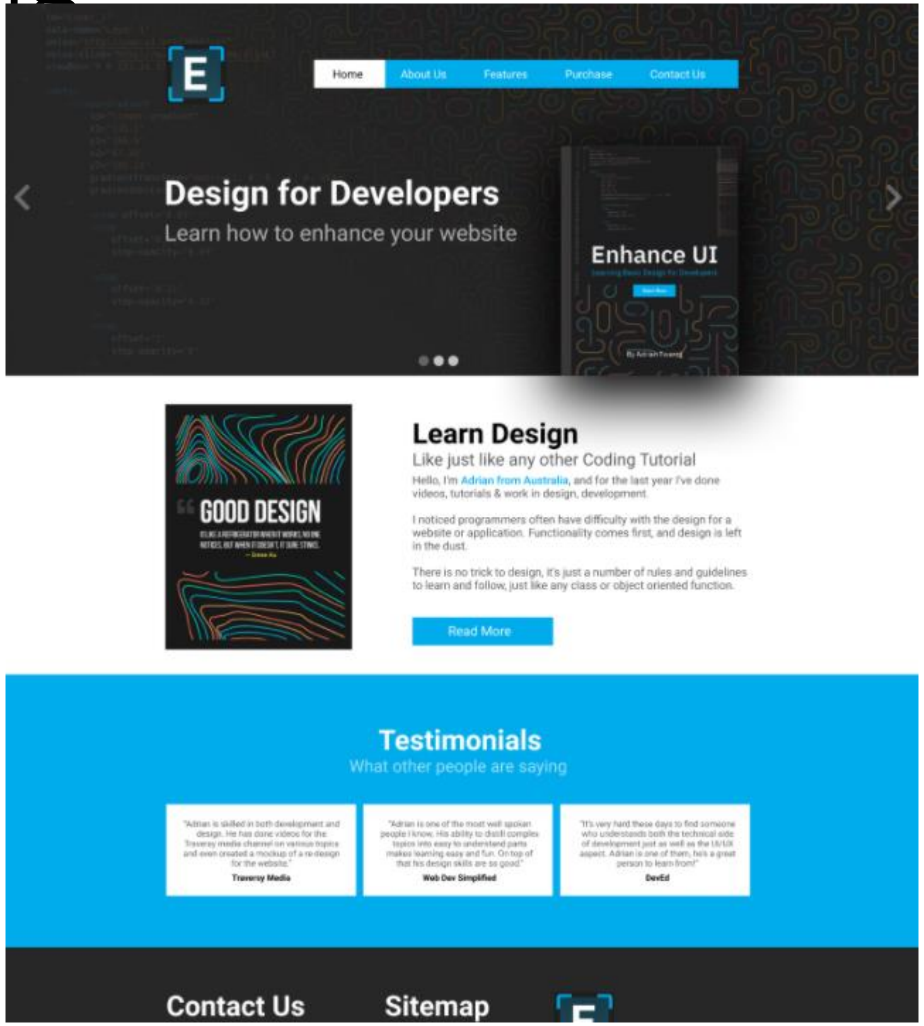
Benefits of low-fidelity

- Less time to create
- Make changes on the fly even during testing with user
    - Navigation doesn’t have to be correct.
- Less pressure on user
- Designer is less committed and can change approach
- Stakeholders know this is an early work that won’t ship soon.

- https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity/

High fidelity UI/UX Prototyping

- Logo
- Colour instead of grey – colour palette
- Images
- Content instead of placeholders
- Hero section – background image.
- Clickable
- Navigation
- Comment and iterate



Design for Developers

Learn how to enhance your website

Learn Design

Like just like any other Coding Tutorial

Hello, I'm Adrian from Australia, and for the last year I've done videos, tutorials & work in design, development.

I noticed programmers often have difficulty with the design for a website or application. Functionality comes first, and design is left in the dust.

There is no trick to design, it's just a number of rules and guidelines to learn and follow, just like any class or object oriented function.

Testimonials

What other people are saying

"Adrian is skilled in both development and design. He has done videos for the Traversy media channel on various topics and even created a mockup of a re-design for the website." **Traversy Media**

"Adrian is one of the most well spoken people I know. His ability to distill complex topics into easy to understand parts makes learning easy and fun. On top of that his design skills are so good." **Web Dev Simplified**

"It's very hard these days to find someone who understands both the technical side of development just as well as the UI/UX aspect. Adrian is one of them, he's a great person to learn from!" **DevEd**

https://www.freecodecamp.org/news/ui-ux-design-tutorial-from-zero-to-hero-with-wireframe-prototype-figma/

High fidelity prototype

- Allows you to test:
  - UI components
  - Graphical elements
  - Page hierarchy
  - Type legibility
  - Image quality
  - Engagement
  - Focus on the test rather than worrying about if the prototype works
  - https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity/

Further considerations

- Are you using a **conceptual model**? Desktop metaphor
- What is your **interaction type** – instructing, conversing, manipulation, exploring or responding.
- What **functions** will the product do?
- How are the **functions related** to each other
    - Sequential or parallel
    - Categorisation (layering/grouping) – privacy on a smart phone

What **information** is needed?

- Data
- How is data to be transformed?

Further considerations cont.

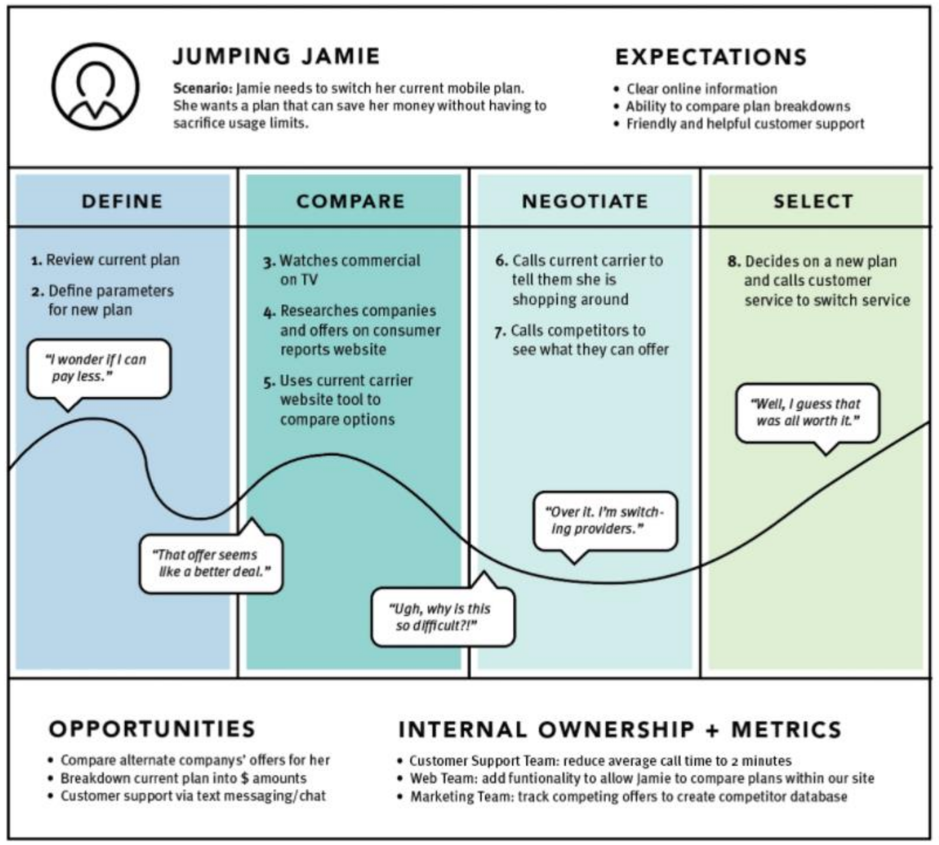
- **Color, icons, buttons, interaction devices**
- **User characteristics and context**
  - Inclusiveness, input and output mechanisms
- **Accessibility**
  - Web content accessibility guidelines
- **Cross-cultural design**
  - Language, colours, icons, information
  - Indigenous knowledge and perspectives

After prototyping go back and explore the users journey

- **Go back to the customer journey map and iterate design.**

1. Persona
2. Scenario and expectations
3. Journey phases
4. Actions, mindsets, emotions,
5. Opportunities

CUSTOMER JOURNEY MAP *Example (Switching Mobile Plans)*



JUMPING JAMIE

Scenario: Jamie needs to switch her current mobile plan. She wants a plan that can save her money without having to sacrifice usage limits.

EXPECTATIONS

- Clear online information
- Ability to compare plan breakdowns
- Friendly and helpful customer support

DEFINE

1. Review current plan
2. Define parameters for new plan

COMPARE

3. Watches commercial on TV
4. Researches companies and offers on consumer reports website
5. Uses current carrier website tool to compare options

NEGOTIATE

6. Calls current carrier to tell them she is shopping around
7. Calls competitors to see what they can offer

SELECT

8. Decides on a new plan and calls customer service to switch service

"I wonder if I can pay less."
"That offer seems like a better deal."
"Ugh, why is this so difficult?!"
"Over it. I'm switch-ing providers."
"Well, I guess that was all worth it."

OPPORTUNITIES

- Compare alternate companys' offers for her
- Breakdown current plan into $ amounts
- Customer support via text messaging/chat

INTERNAL OWNERSHIP + METRICS

- Customer Support Team: reduce average call time to 2 minutes
- Web Team: add funtionality to allow Jamie to compare plans within our site
- Marketing Team: track competing offers to create competitor database

https://www.nngroup.com/articles/journey-mapping-101/

Norman Nielsen Group (1998)

- **Norman Nielsen Group** https://www.nngroup.com/
- **Don Norman** – father of UX
  - Coined term UX
  - Professor, researcher in design, usability, cognitive science, user-centric design
  - Apple – "User Experience Architect" in early 90s
  - Book - *"The Design of Everyday Things*"
  - Internet interactions – patents
- **Jakob Nielsen**
  - Creator of usability heuristics

10 Usability Heuristics for User Interface Design (Jakob Nielsen)

1. Visibility of system status
2. Match between the system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency
8. Aesthetic and minimalistic design
9. Help users recognise, diagnose and recover from errors
10. Help and documentation

https://www.nngroup.com/articles/ten-usability-heuristics/

Next : Workshop 03

- Ideate functionality i.e. solutions to “How might we… ” problem statement
- Functional and Non-Functional Requirements
- Start wireframe/lo-fi prototypes
- Competitor analysis – best in World