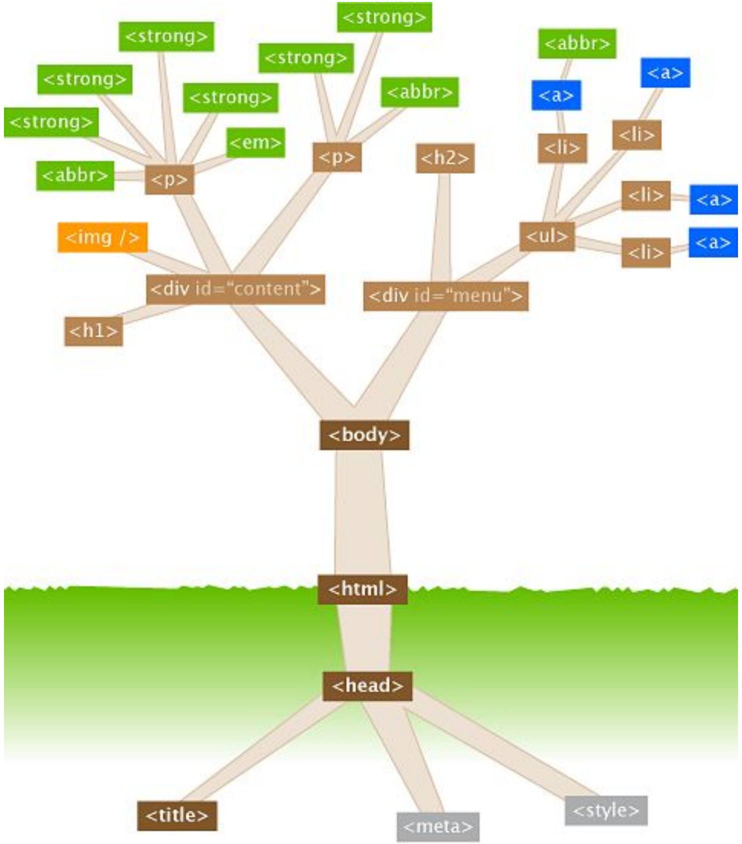
WEP – UI Programming basics

Agenda

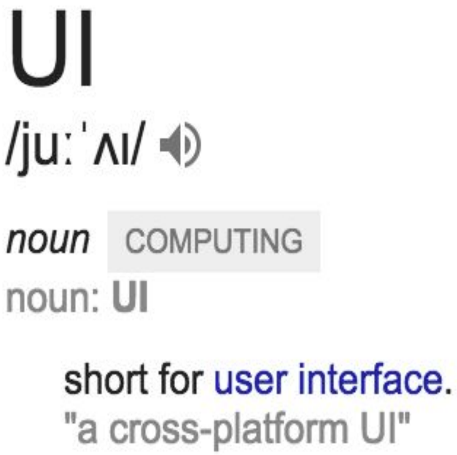
- What is a UI?
- What is state?
- What is model?
- Programming blocks



What is a UI?

where a user with the program logic interacts

- the interface with which the user interacts.
- Through a UI, the user can actuate actions
- UI can throw event
- UI can react to other events
- UI is the interface between human and system



UI

/juːˈʌɪ/ 🔊

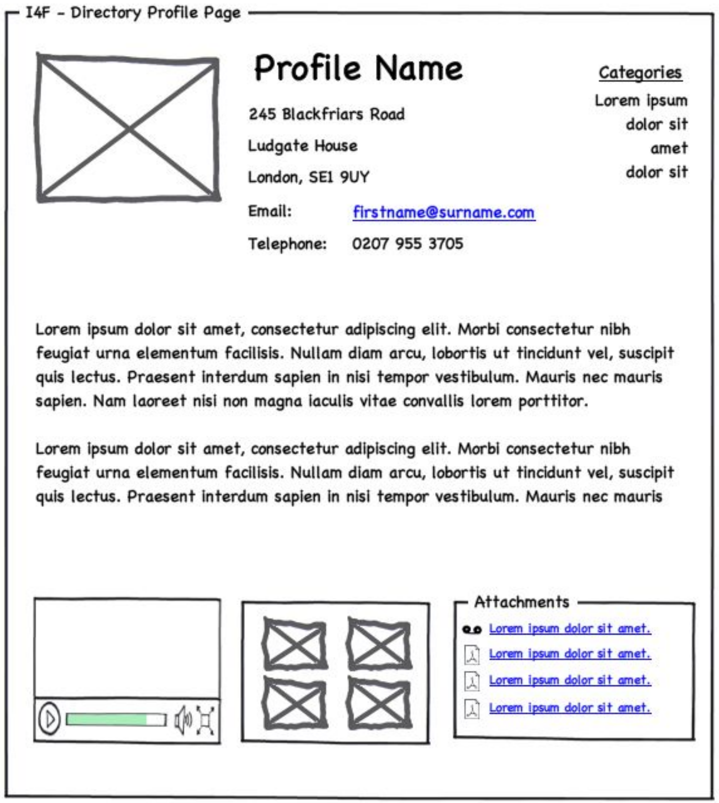
noun COMPUTING

noun: **UI**

short for user interface.

"a cross-platform UI"

User Interface



I4F - Directory Profile Page

Profile Name

245 Blackfriars Road
Ludgate House
London, SE1 9UY

Email: firstname@surname.com

Telephone: 0207 955 3705

Categories

Lorem ipsum
dolor sit
amet
dolor sit

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi consectetur nibh feugiat urna elementum facilisis. Nullam diam arcu, lobortis ut tincidunt vel, suscipit quis lectus. Praesent interdum sapien in nisi tempor vestibulum. Mauris nec mauris sapien. Nam laoreet nisi non magna iaculis vitae convallis lorem porttitor.

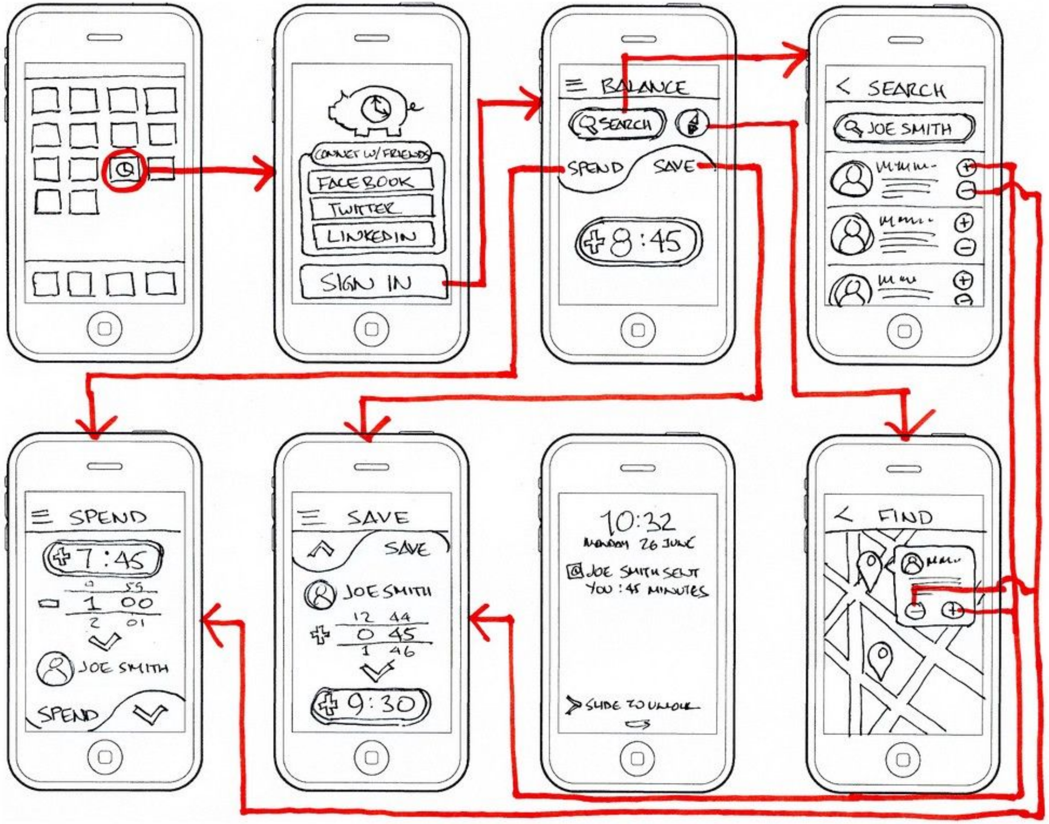
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi consectetur nibh feugiat urna elementum facilisis. Nullam diam arcu, lobortis ut tincidunt vel, suscipit quis lectus. Praesent interdum sapien in nisi tempor vestibulum. Mauris nec mauris

Attachments

- Lorem ipsum dolor sit amet.
- Lorem ipsum dolor sit amet.
- Lorem ipsum dolor sit amet.
- Lorem ipsum dolor sit amet.

created with Balsamiq Mockups - www.balsamiq.com

Wireframes

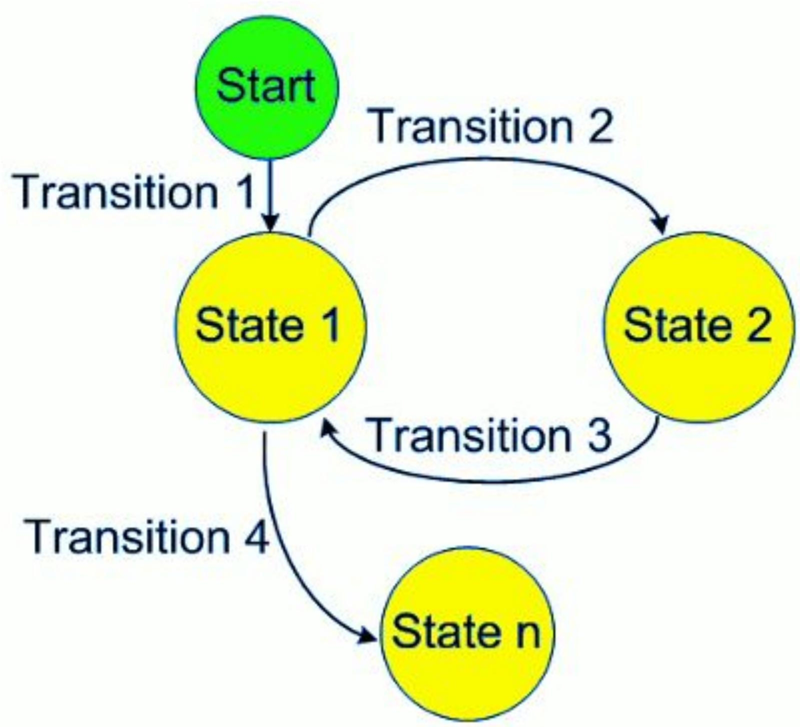


Wireframes with the Flow

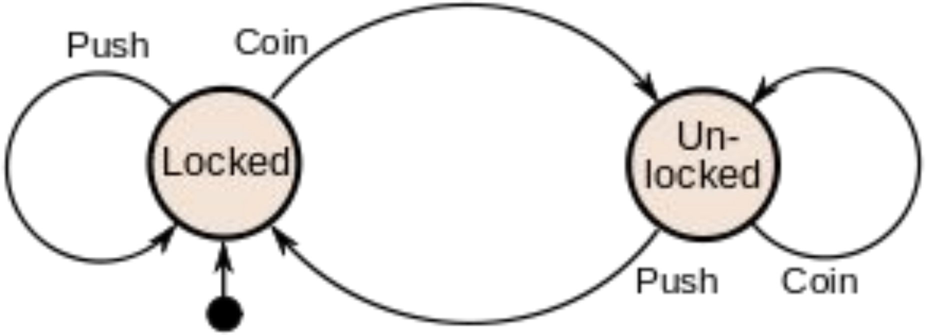
Whats is a state?

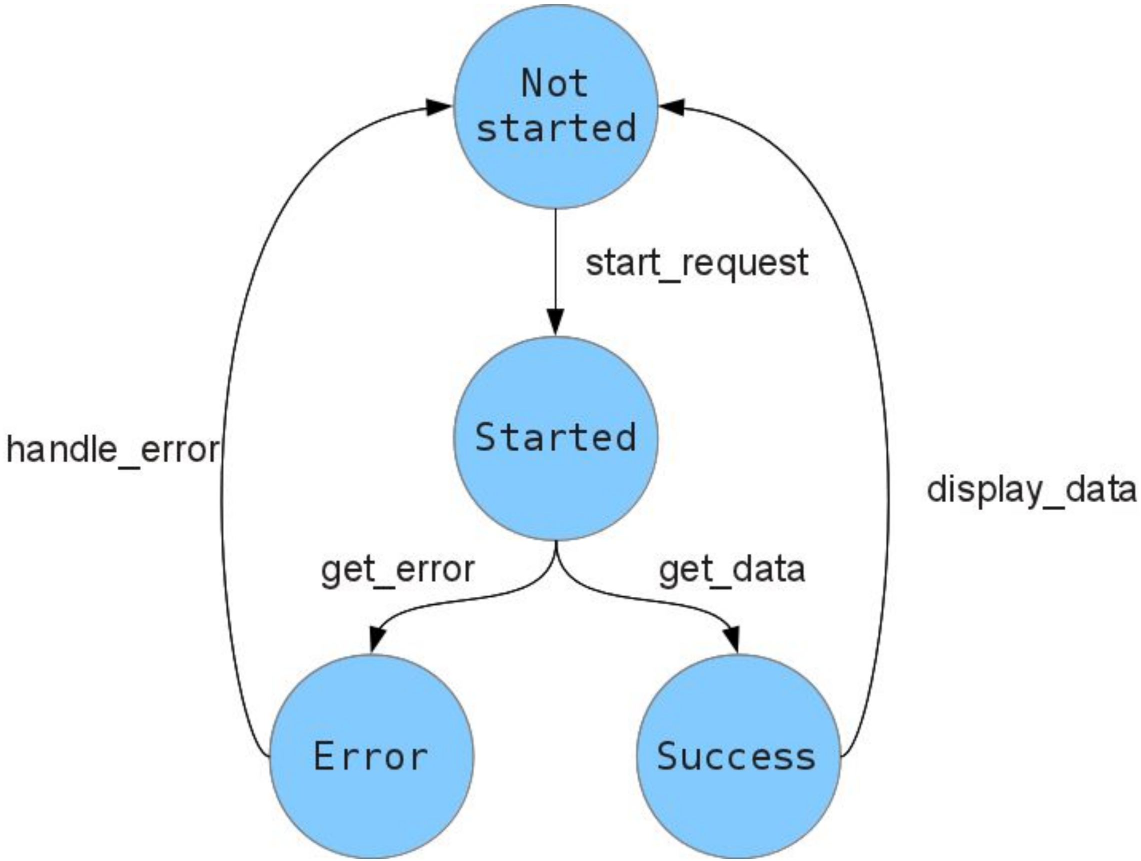
snapshot of a current program situation

- The situation at a given time
- In programming, the information about the context at a particular moment.
- Values of variables, arrays, objects, etc…
- The values of a programming moment



States and Transitions


Push
Coin
Locked
Un-
locked
Push
Coin

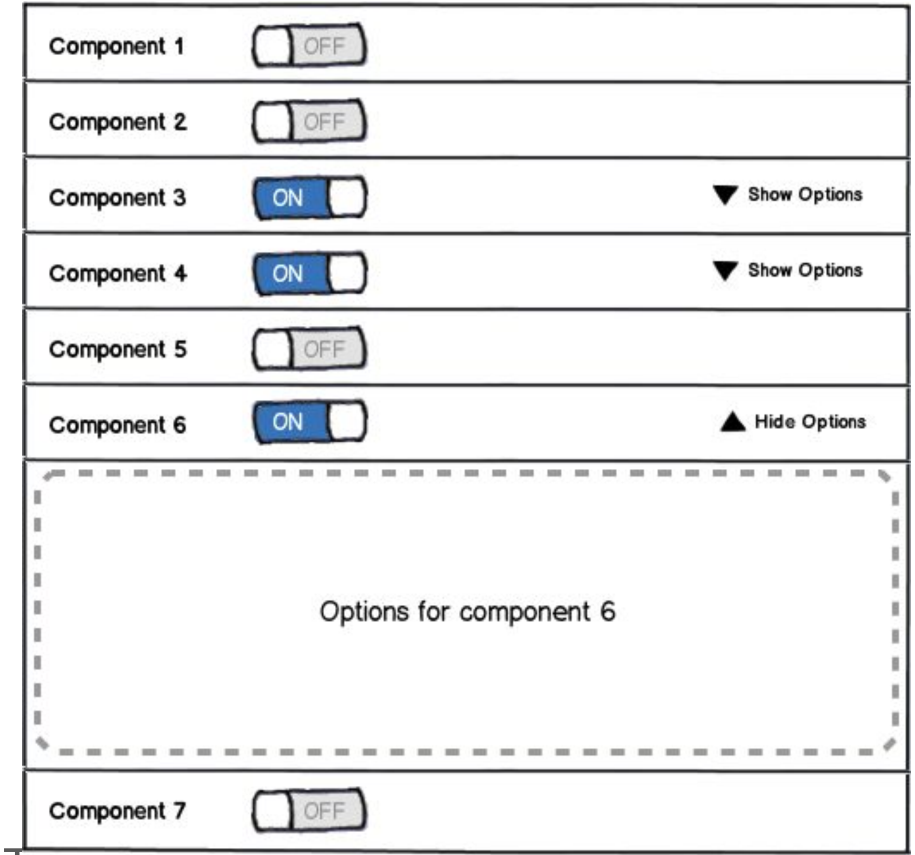


Ein Beispiel

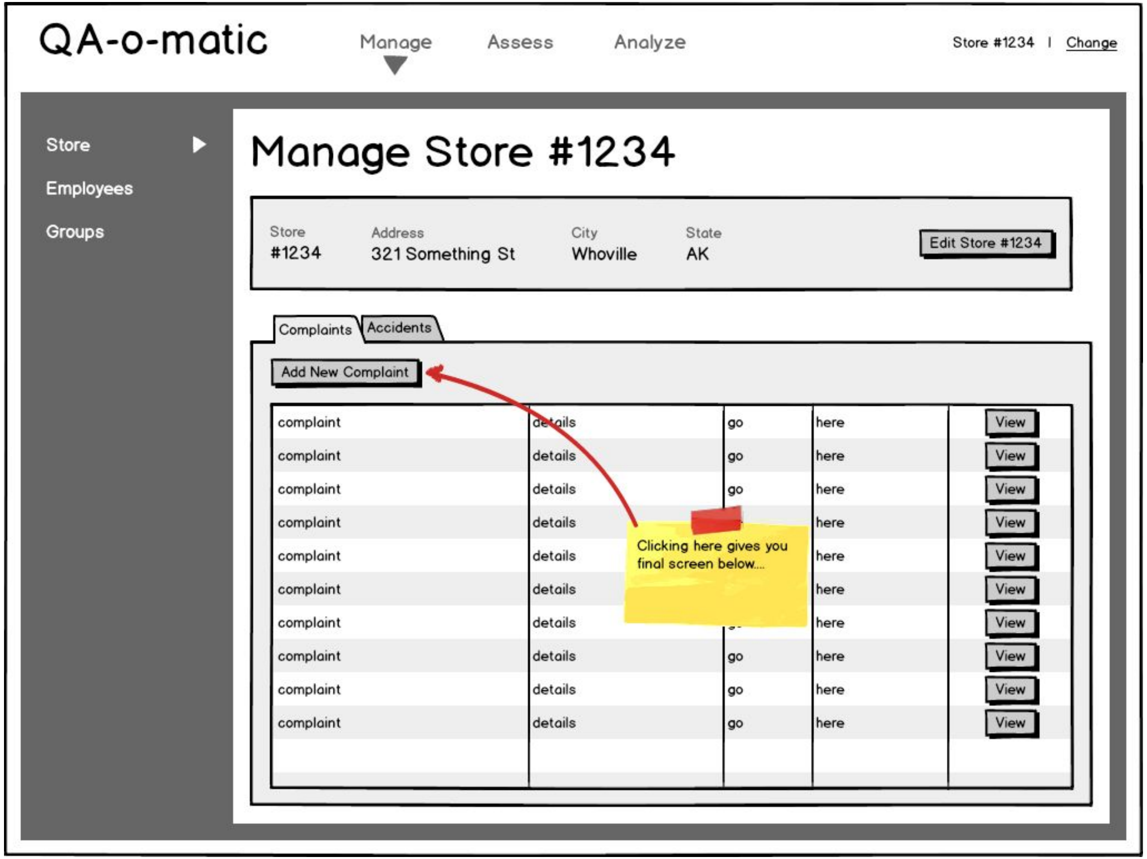
UI State

Where to store ui state information

- JS is responsible to read and manipulate DOM
- No state kept in UI
- States are kept in Javascript with the Javascript structures
- Use
    - Variables
    - Arrays
    - Objects
    - Class
    - Data structures (Stack, Queu, Linkedlist, etc...)
- If you are sure that your state has been updated, the update UI.



The state of the UI: Three options are active!
So three variables (boolean) in the program have the value “true”.

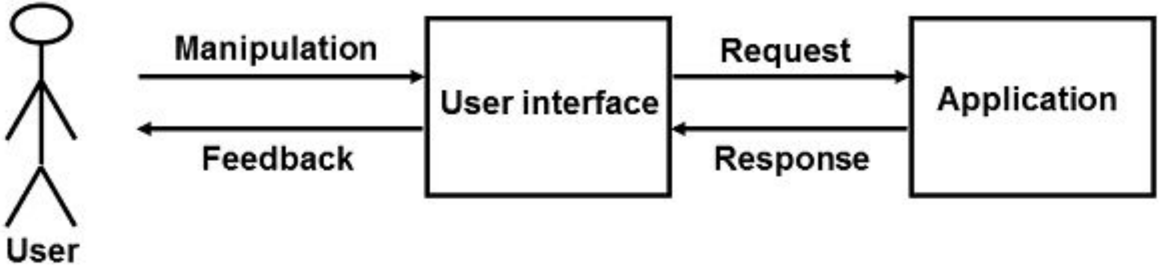


Manage Store #1234

Store	Address	City	State
#1234	321 Something St	Whoville	AK

complaint	details	go	here	View
complaint	details	go	here	View
complaint	details	go	here	View
complaint	details		here	View
complaint	details		here	View
complaint	details		here	View
complaint	details		here	View
complaint	details	go	here	View
complaint	details	go	here	View
complaint	details	go	here	View

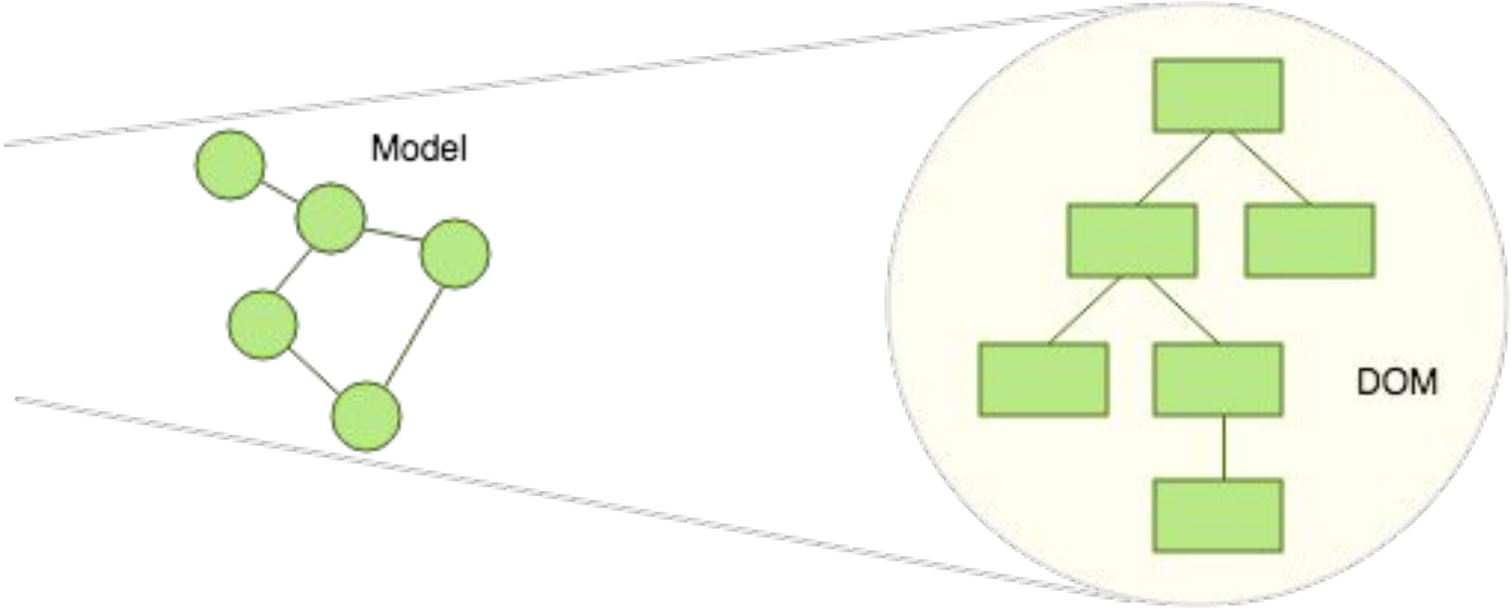
Clicking here gives you final screen below....


User
Manipulation
Feedback
User interface
Request
Response
Application

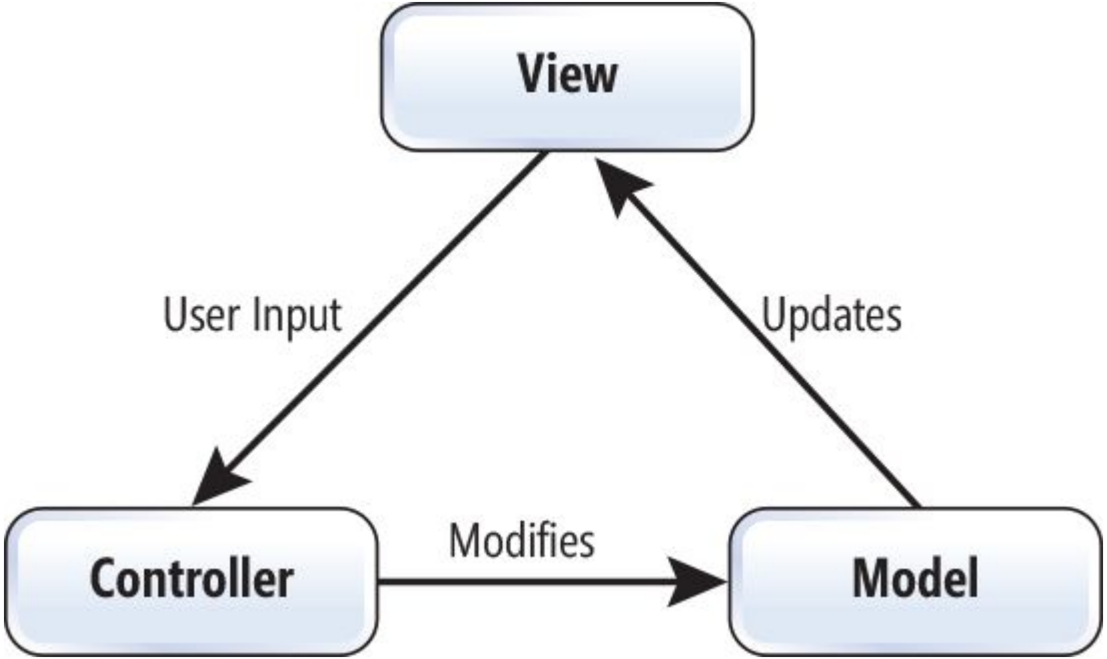
The model

A data structure to represent your program logic

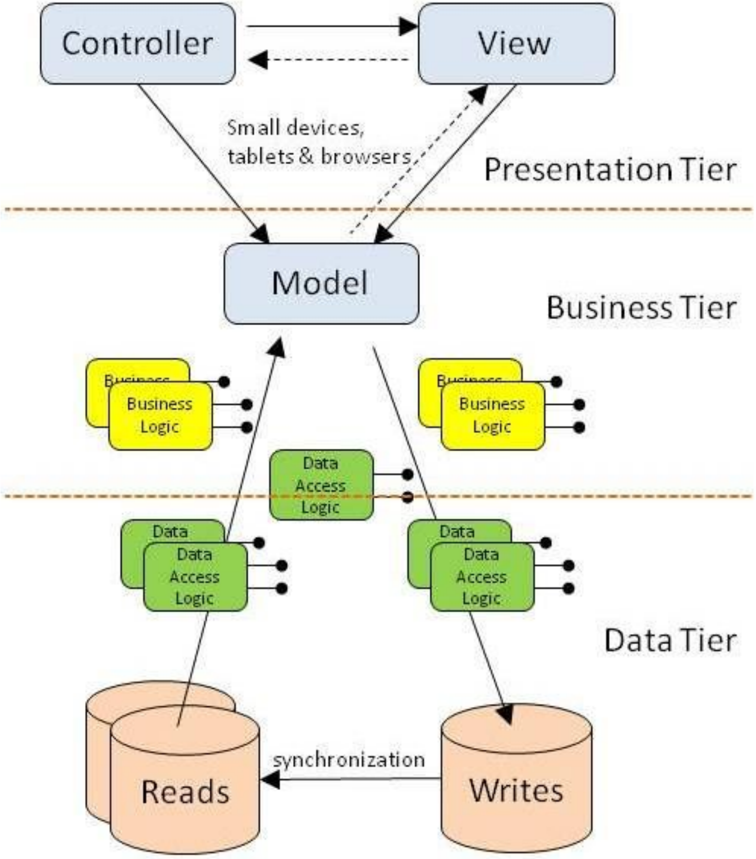
- Each program has to have a model
- It is representing the how the related data is being kept in the course of the program execution
- Single variables are the most basic model elements
- Arrays (list) and objects are the most useful ones in the programming to process huge data at easy.



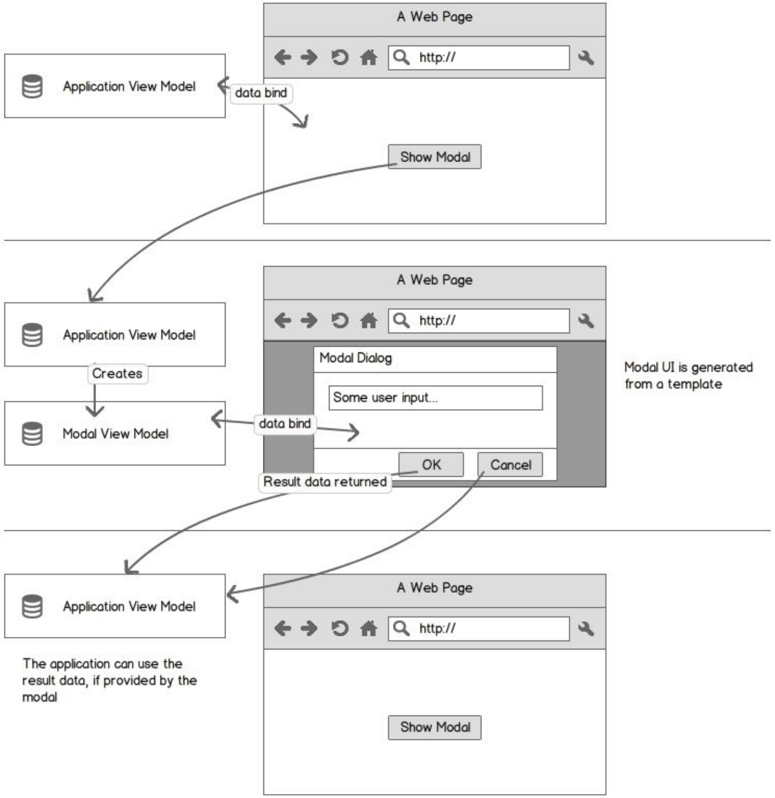
UI Object Modelling mit Javascript Konstrukten



MVC Pattern



Mehr Details



Interaction with Model

UI Programming approach

Basic building blocks in UI Programming

- Create your model based on your analysis
  - Do you want to use variables, arrays, objects or any combination of them
  - Try to put ui state in a model (data structure). An object may cover your needs, where possible.
- Try to separate the general logic and the functions from each other
  - it helps you to have an overview and manageable code base.
  - You can event separate the functions into many files, based on their relationships such as functions updating DOM, functions changing some part of the data, functions creating statistics, etc…
  - You can even put event listeners in a separate file.
- Start with a static html
  - You can just first start with a static html, how you dynamic UI would look like. And then take the html apart and finally create the functions representing those parts
  - Use minimal number of container elements in the static html, where the dynamic contents are to be injected.

Basic building blocks in UI Programming

- Use arrays methods, where ever possible
  - to filter a list
  - the convert the model
  - to generate dynamic html
  - to count the number of something in the model
  - to find a total
  - to sort a list
- You may need to use objects and objects methods
  - to handle special cases like counting values
- You may need to use sets and maps
  - to find unique values in a list
- Use the functions excessively, to create the most meaningful, smallest parts of the program
- Use template literals `hi ${firstname}`

Basic building blocks in UI Programming

- Use eventlisteners wisely
    - if the content is being changed often then do not use any event listener in this content, instead set the listener on the parent, which probably is not dynamically updated.
-

Create a model

here we go:

```
let productCount = 0;
```

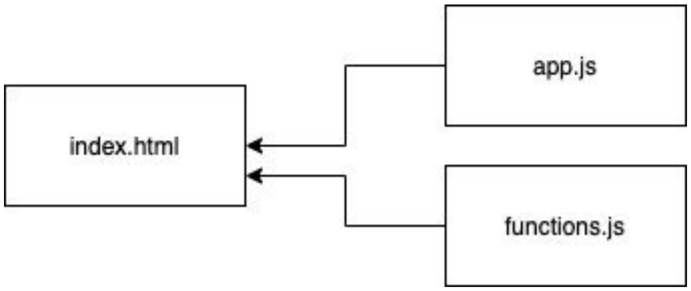
a complex one:

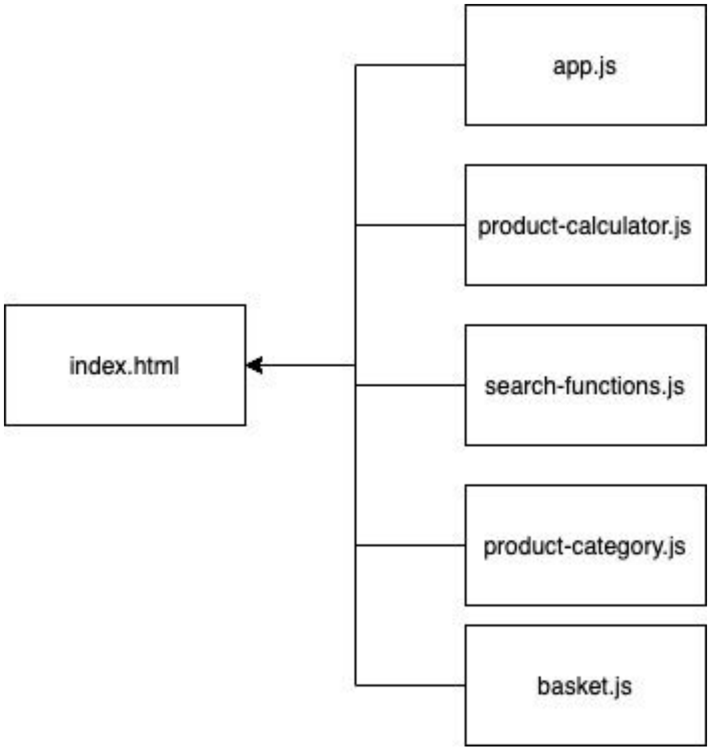
```
let person = {name:”ali”, children: [“ayse”, “ahmet”], isWorking: true,
image:”data:image/png;hjwdhjwhjdw==”}
```

and another one:

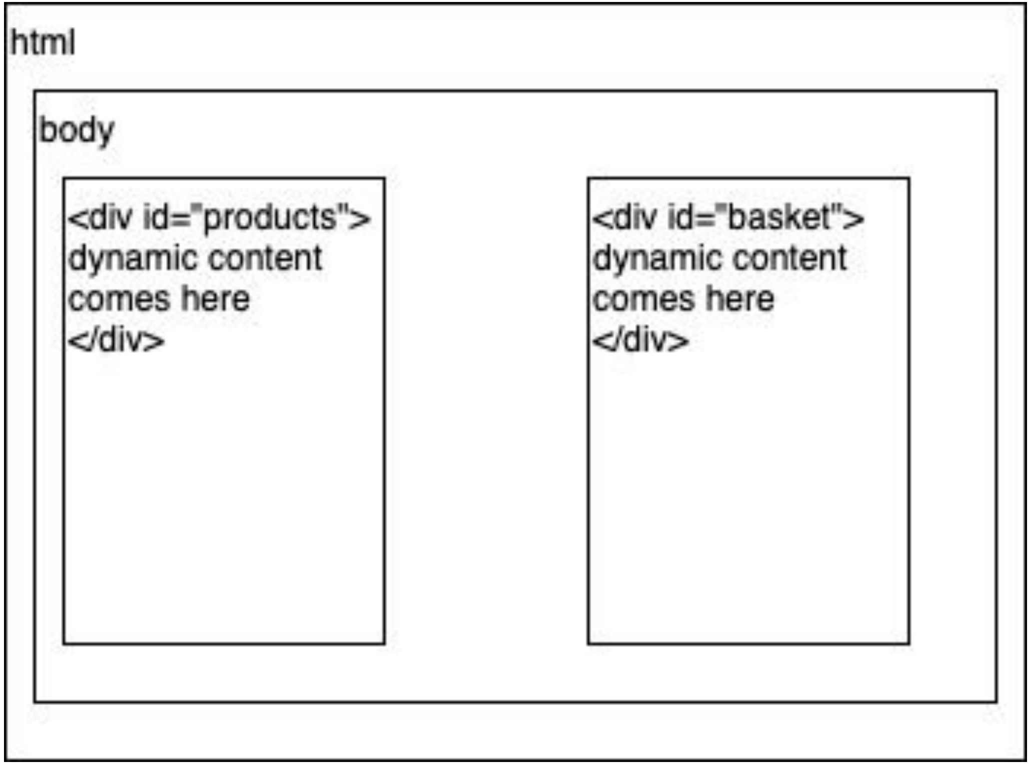
```
let gameConsole = {isRedLightOn: true, isGreenLightOn: false, playerPosition: [4, 5]
isGameFinished: false}
```

Multiple separated files





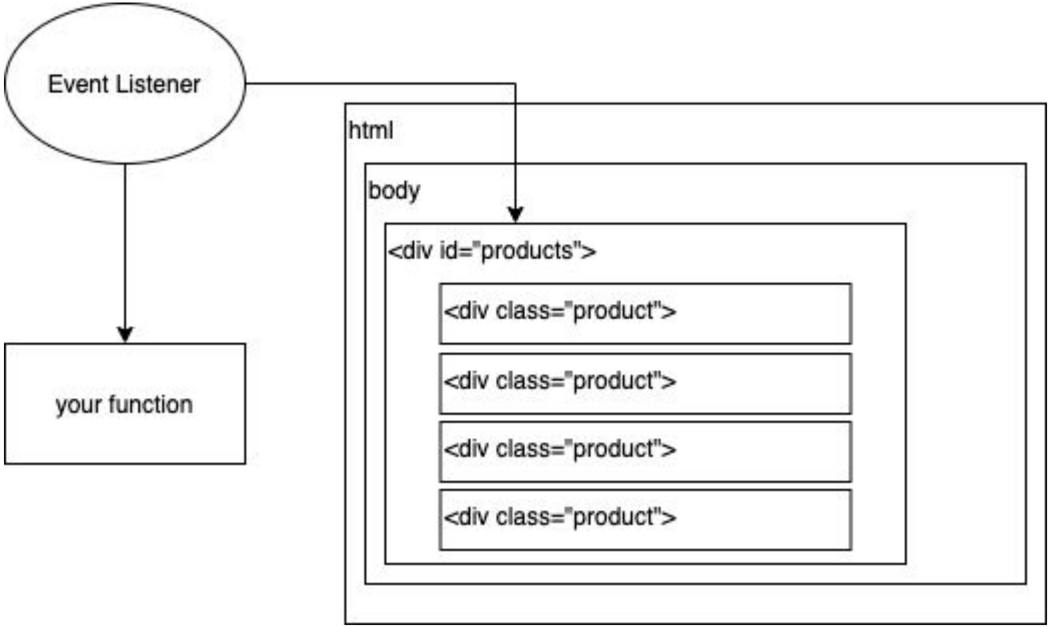
Create your static content with minimal html



```
<div id="products">
dynamic content
comes here
</div>
```

```
<div id="basket">
dynamic content
comes here
</div>
```

Put your listeners on the least dynamically changed parent



let's try

Questions?