UML

Person —general→ Student

Animal —Realize- - -▷ Dog

Application —◇ Aggregation → Letter

Nurse —Dependency→ Chart

Three Tier Architecture

Get input from user. Display output.

Perform calculations. Make decisions.

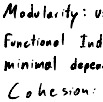
Store and retrieve data.

Understandability, Stability, Changeability, Testability.

Software Design Principles

- Modularity: use subsystems
- Functional Independence: minimal dependence between files.
- Cohesion: the degree to which elements inside a module belong together.

MVC



View

Controller

Model

DOM

- allows programs to dynamically update content/style of documents.
- feature of the web browser.
- The DOM allows JS to:
  - change HTML elements in the page.
  - change CSS styles.
  - remove existing HTML elements.
  - react / create HTML events.
- Node Abstract type Operations:
  - parentNode: returns the parent.
  - parentElement: returns the element of ↑
  - hasChildNodes(): True/False
  - childNodes: return children
  - firstChild / lastChild.
  - previousSibling / nextSibling.
- Document Abstract type:
  - getElementsByTagName();
  - getElementsByClassName();
  - createElement(localName);
- Element Abstract type:
  - id, className
  - getAttributeNames()
  - getAttribute(qualifiedName)
  - setAttribute(qualifiedName)
  - removeAttribute(qualifiedName)
- Event object
  - type "click", "submit".
  - target event's target.
- EventTarget object
  - addEventListener(type, callback)
  - removeEventListener(- II -)
  - dispatchEvent(event)
- DOM Manipulations
  - let h = document.getElementById("h1")
  - h.innerHTML = "Welcome";
  - h.style.backgroundColor = "yellow".

Rest API

Stateless interaction: Each request contains all necessary info.

Defines: system components, their interactions, interface protocols, communication per interaction.

Core concepts: Resources, their representation, communication over HTTP, web view app state.

Resources: any data manipulated by App, must have ID, HTTP req must identify a resource.

Representation: resource might have multiple reps, server returns resource in specified format.

How does the REST API work?

Client sends a formatted API request, server authenticates client and verifies their rights, server processes request and sends response.

Benefits of Rest API

Scalability: Statelessness reduces server load, no need to retain past client info.

Flexibility: decoupling components allows them to evolve independently.

Independence: You can write client and server apps in various languages.

Design of REST API

Identify resources and their attributes, give resources unique names, map each resource to the relevant operations, define endpoints and HTTP requests.

UI Design

User Flow diagrams are charts that show how a user takes actions through an application to go from one step to another.

Wireframes give the designer the opportunity to quickly iterate on many ideas, receive feedback on technical feasibility, usability.

Prototypes helps with usability testing and getting user flows together.

Helpful for development because they are rep representations of final product.

HTTP: network protocol of the web.

Protocol: system of rules allowing entities to transmit info.

IP address: numerical label assigned to each device using Internet.

Port: Communication endpoint identified by 16-bit unassigned int.

WWW: information system where resources are identified by URL.

Transactions consist of: initial line, header line fields, blank line, optional msg.

GET: request representation of resource. PUT: update or replace. DELETE

POST: send data to the server, create a new resource on the server.

Initial line response: HTTP_version status code.

Initial line request: Method (GET) URI (/index.html) HTTP_version

Header line request: From: provides email of entity making request.

User Agent: Program-name/x.xx. Mozilla/5.0.

Header line response: Server: server: product. server: Amazon S3.

Expires: http-date timestamp. Thu, 13 Nov 1981 ...

Message body: Content-Type: gives the MIME-type of the data in the body.

Content-Length: gives the number of bytes in the body.

Curl: curl -X POST -d "name=KO & major=CIS" http://theserver.edu

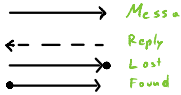
curl -X DELETE http://theserver.edu/user/19.

Sequence Diagrams

focuses on message interchange between lifelines

Message: named element that defines one specific kind of communication between lifelines of an interaction. i.e function call. HTTP request.

Lifeline: named element that represents a participant in interaction.



Web Development

Rendering: The browser parses HTML/CSS files into the DOM. The DOM combines the documents content with its style.

Render tree: Start at the root of the DOM tree, traverse each node.

Ignore invisible nodes (script tags) and hidden nodes via CSS. For each visible node find CSSOM rules and apply them. Add visible nodes.

Attribute Selectors

[attribute]: selects all elements containing that attribute. a[href="https..."]

[attribute="val"]: - II - containing that attribute, but only if the value is val.

[attribute~="val"]: - II - containing that attribute and val in its value.

Combinators

Adjacent Sibling: former element + target element {style properties}.

General Sibling: former element ~ target element {style properties}.

Child: former element > target element {style properties}.

Descendant: former element target element {style properties}.

!important assigned after a property assigns the highest priority.

Hierarchy ranks selectors by how many elements they match.

Ranking: Classes and Attributes, IDs, Inline styles

Specificity: +1 for elements, +10 for each class, +100 for ID, +1000 inline

JavaScript

First class functions language, supports OO, functional programming (imperative)

Dynamic typing: type checking at runtime. case sensitive.

Hoisting: variable declarations moved at the top of their scope.

Primitive types are immutable. null is type object, undefined undefined.

Destructuring arrays/objects: [r, e, ...rest]. works for objects

Exceptions: throw new Error("No value was entered");

Functional Programming and JS

functions can be stored in a variable, passed as input to other functions.

Anonymous Functions: functions declared without a named identifier.

Arrow Functions: (e) => {e.target.id} braces not needed if single exp

Pure Functions: Function with no side effects no input mutation

HOF MAP: executes the input function on every element of array and return a copy containing the results. Original array not modified

```
[1,2,3,4].map((x) => x*2)  => [2,4,6,8]
```

HOF Reduce: reduces the array to a single value

```
const reducer = (accumulator, current) => accumulator + current;
[1,2,3,4].reduce(reducer, 0)   (0 = base)   => 10.
```

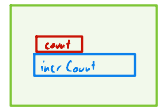
HOF Filter: creates a new array of entries that pass the test

```
[1,2,3,4].filter((x) => x%2 == 0)  => [2,4]
```

Closures and Scopes: used in React to manage states

```
function makeCounter(){
  let count = 0;
  function incrCount(){
    count = count + 1;
    console.log(count);
  }
  return incrCount;
}
```

Heap



OOP in JS: prototype oriented language

No explicit classes, Objects augmented to create new ones. An object is a collection of properties with a single prototype obj.

```
function Student(name, gpa){
  this.name = name;
  this.gpa = gpa;
}

const s1 = new Student("Jamal", 3.6);
Student.prototype.getInfo = function(){return this.name;};
s1.getInfo(); => "Jamal"
```

The prototype property can be used to extend an object.

call() invokes an internal method of a function obj.

```
function UGStudent(name, gpa, major){
  Student.call(this, name, gpa);   → invoke Student's constructor function
  this.major = major;
}
UGStudent.prototype = Student;
let y = UGStudent("klejdi", 4.0, "CIS");
y.getInfo() → "klejdi"   inheritance
```

React: reusable and extensible components

npx create-react-app <name_of_app> npm start

package.json: allows npm to manage the project's dependencies.

Host Tree: React outputs a tree that changes in response to events. A renderer allows React to interact w/tree.

ReactDOM renders a react tree into host's DOM tree.

React Element: Javascript object that defines a node on DOM tree. Attributes: type (HTML element) and props.

JSX: syntax extension to JS that produces React elems.

React Elements are immutable. Destroyed when tree changes.

Component: A function that returns React elements.

Takes props as argument and uses state to update.

Components cannot update their props → pure.

Virtual DOM: copy of the DOM kept in memory. When updating, ReactDOM synchronizes the Virtual DOM and the real DOM through reconciliation.

Babel: compiles React code into Javascript.

Webpack: Static module bundler for JS applications