Software Development 2

Session 5

Today

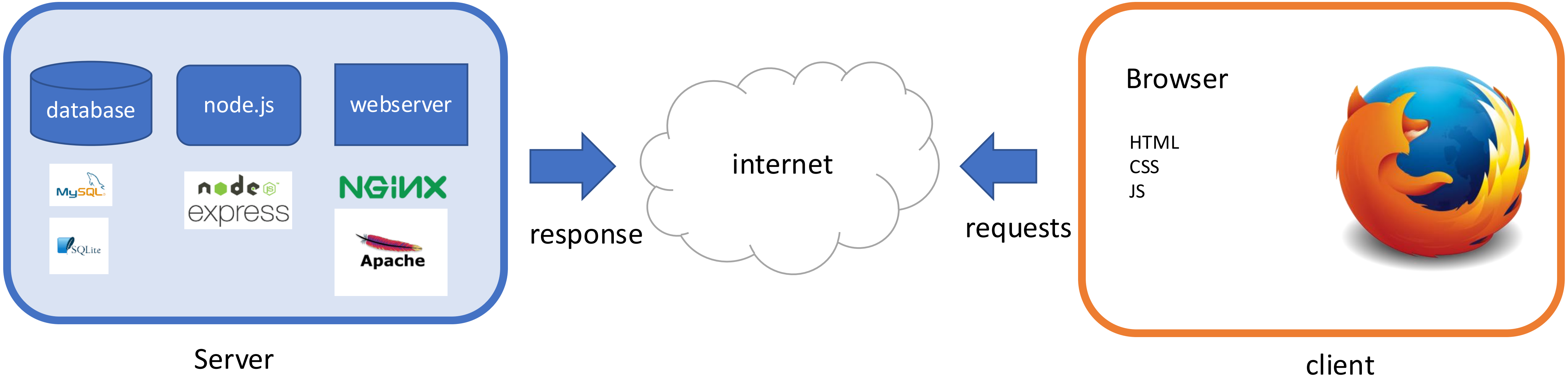
Lab

- Setting up docker/node/express/mysql
- Checking connectivity to the database
- Review Express routing exercises from last week

Seminar

- Sprint 2 reviews
- Database design (ERD etc)

Client/server software interaction over the internet – this diagram represents how a web application works when you access it as a user


database

node.js

webserver

response

internet

requests

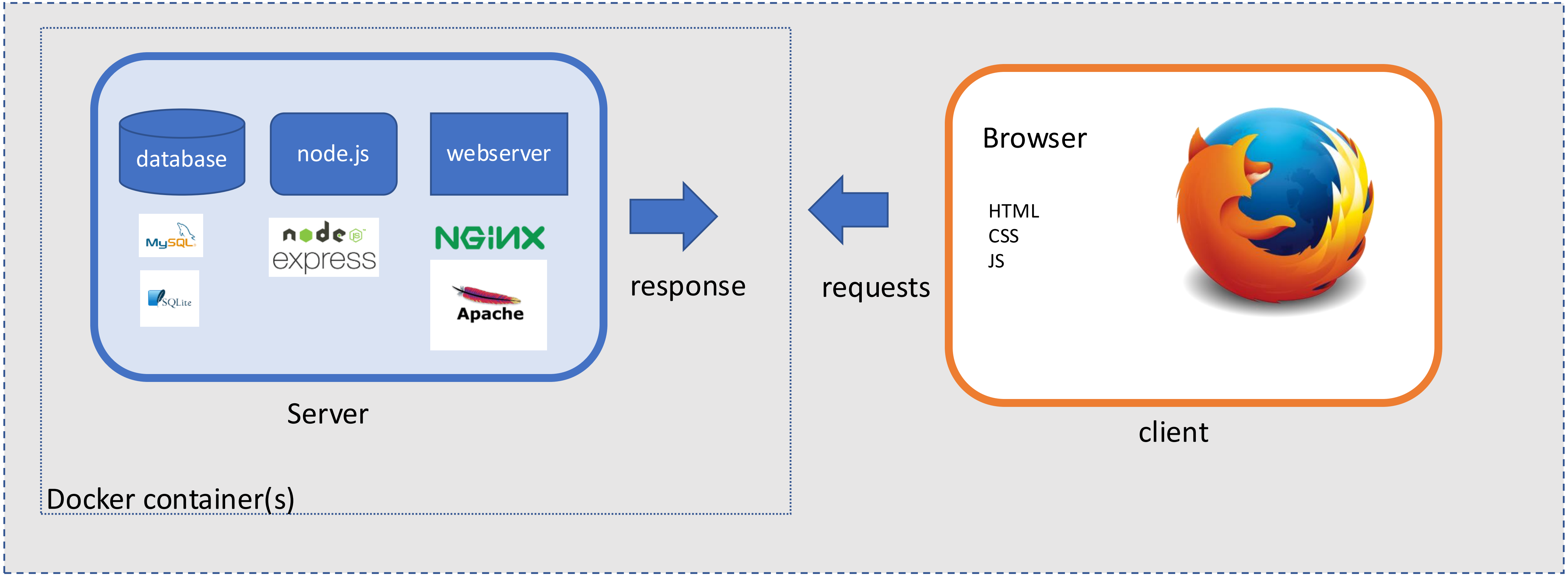
Browser

HTML
CSS
JS

Server

client

Development environments: When projects are in development, developers will have all the client and server software running on their own computer. Additionally they will use tools such as VS Code and Git to write and manage source code.



database

node.js

webserver

Server

response

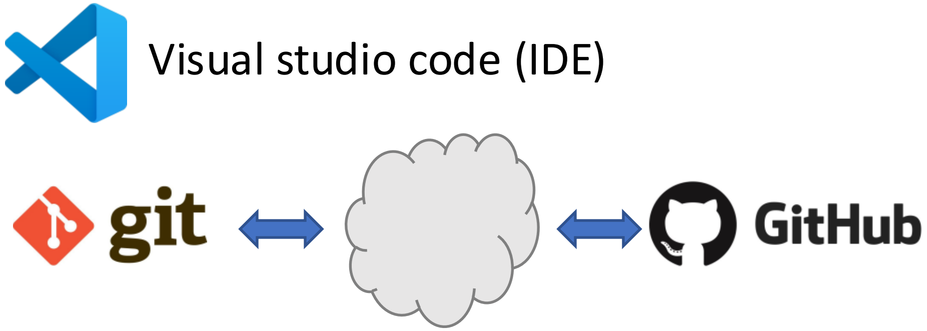
requests

Browser

HTML
CSS
JS

client

Docker container(s)



Visual studio code (IDE)

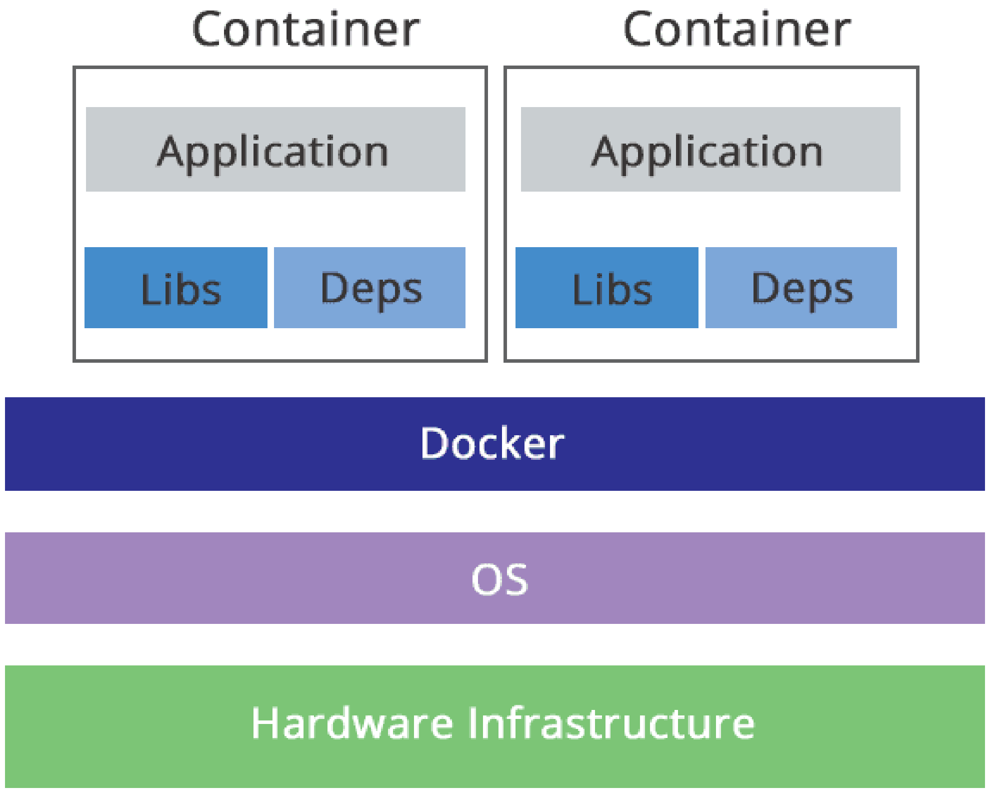
What is docker?

- An industry standard tool for ensuring that development, test and production environments are consistent
- A 'devOps' tool ie. Defines `infrastructure in code` / text based configuration files
- This can then be versioned, exchanged etc to create consistency

DevOps and Agile

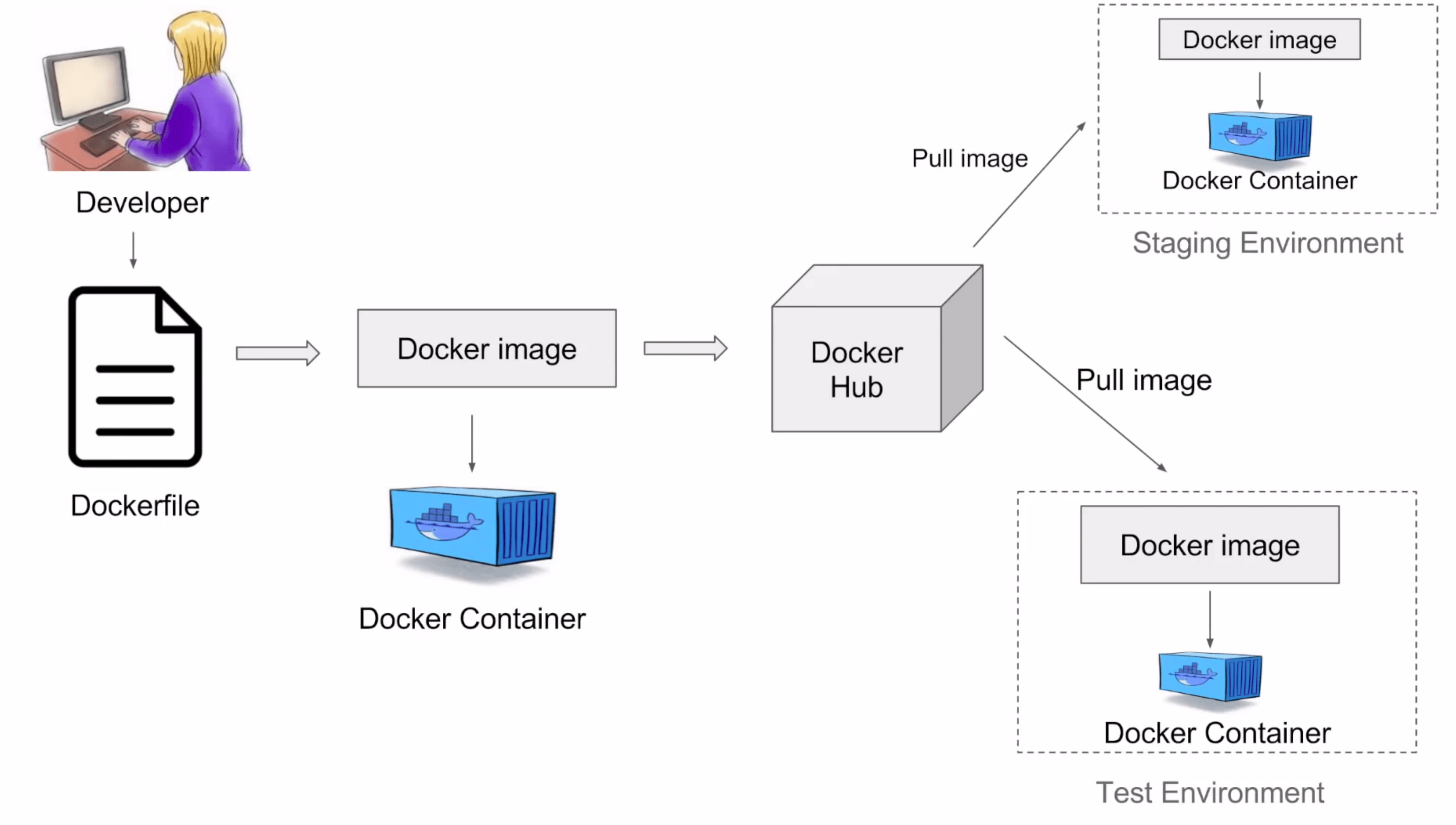
- Tools like Docker are key to CI/CD, as they allow quick and reliable deployments and consistent environments.

- CI – Continuous Integration
- CD – Continuous Delivery



CONTAINERS

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.


Developer
Dockerfile
Docker image
Docker Container
Docker Hub
Pull image
Docker image
Docker Container
Staging Environment
Pull image
Docker image
Docker Container
Test Environment

Getting Docker running on windows on your own laptop

- Download docker desktop: https://www.docker.com/products/docker-desktop/
- Run the wsl commands from the lab sheet
- Install the kernal update package -
- Restart
- Install docker desktop
- Test by running the command:

```
docker run hello-world
```

- restart

Lab today

- Making sure you can run Docker (lab or your own laptop) and node.js
- Downloading and running a set of files which will get you started
- Connecting to the database
- A simple exercise in retrieving values from the database and sending them to the browser.
- Make sure you have completed all the Express routing exercises from last week in the Docker context.

Docker desktop update and check

- update docker desktop
- Images->search images to run
- search for hello-world
- click run
- click 'yes' to run a new container.
- logs/terminal should open saying 'hello from docker'

- Break

- Sprint 1 review meetings

Seminar – Sprint 2

Submission requirements for this Sprint:

- At least two personas
- Full set of user stories defined for your application idea
- User stories tracked appropriately on your task board
- Wireframes for your chosen user stories
- User flow chart or activity diagram for your chosen user stories
- You may also include other diagrams and documentation used to design software, for example:
  - Use case diagrams
  - State diagrams
  - Sequence diagrams
  - Customer Journey maps
- Your wireframes should be rough designs of the screens (pages) that your application will have and please indicate how these flow together. They do not need to be made using a digital tool - photographed sketches are OK. The point is that they should provide a useful specification.

User stories

"*A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.*"

How to write a user story

What to include

Your user stories should include enough information for your product manager to decide how important the story is. They should always include:

- the person using the service (the actor)
- what the user needs the service for (the narrative)
- why the user needs it (the goal)

The right format for user stories

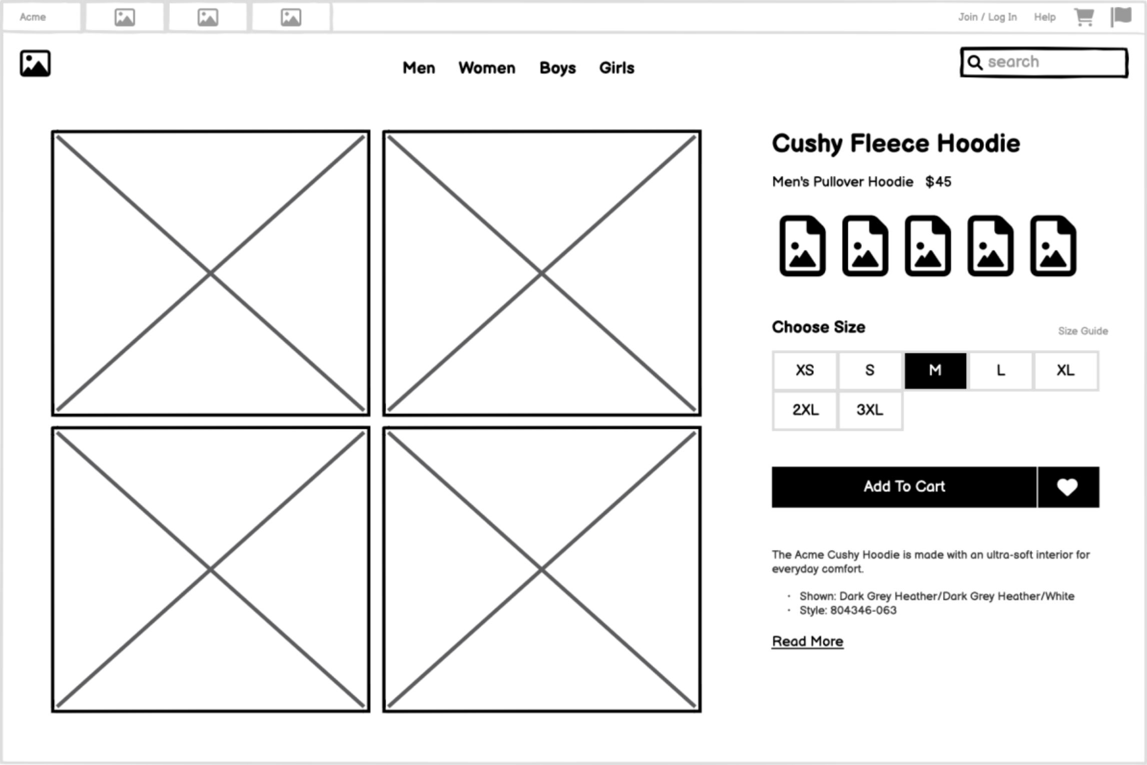
- They’re usually written in the format:
- As a… [who is the user?]
- I need/want/expect to… [what does the user want to do?]
- So that… [why does the user want to do this?]

You can also add 'acceptance criteria'

https://www.gov.uk/service-manual/agile-delivery/writing-user-stories

Wireframes

"A wireframe is a schematic or blueprint that is useful for helping you, your programmers and designers think and communicate about the **structure** of the software or website you're building."

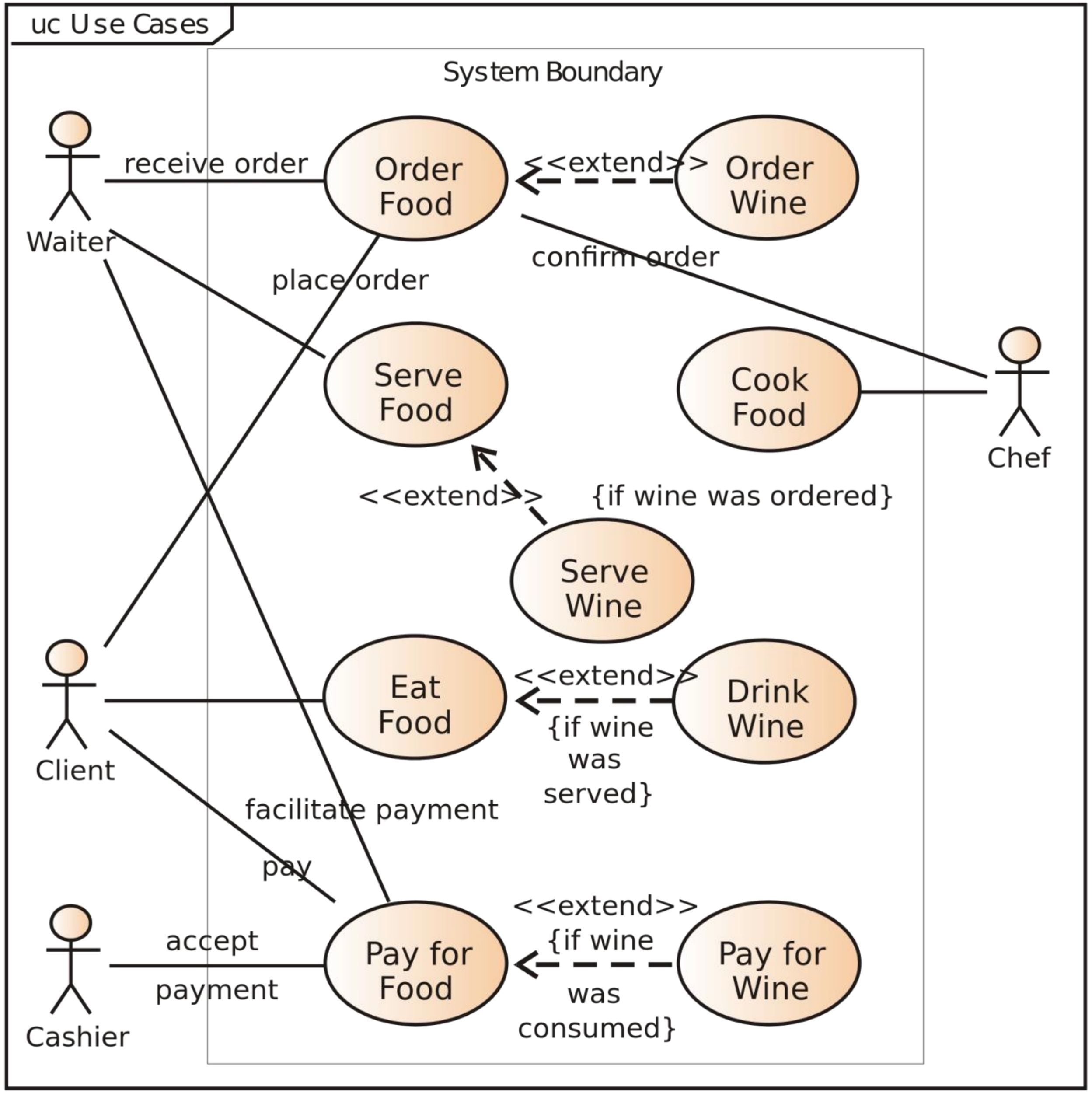


The Acme Cushy Hoodie is made with an ultra-soft interior for everyday comfort.

- Shown: Dark Grey Heather/Dark Grey Heather/White
- Style: 804346-063

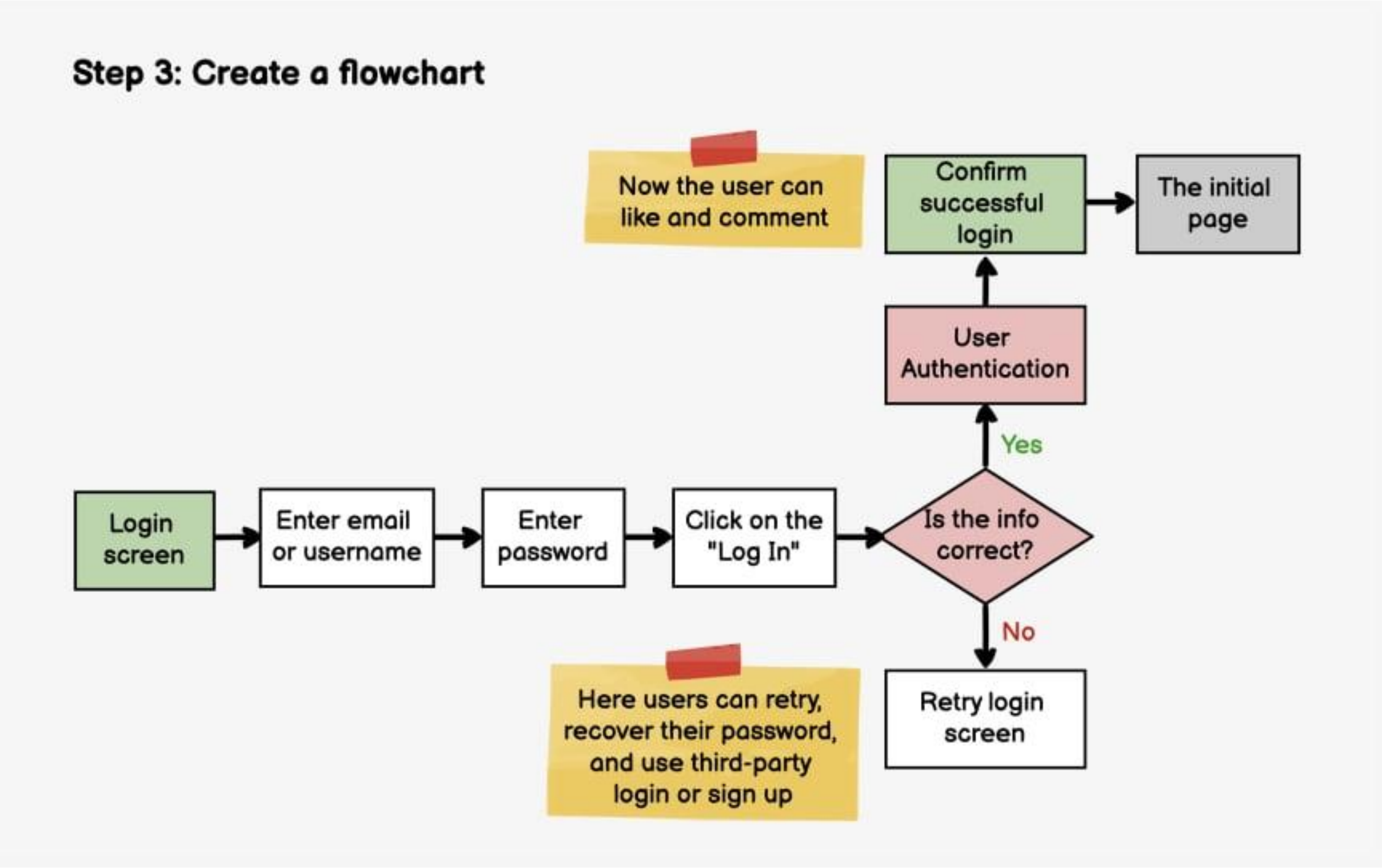
Use-case

Captures how users interact with the system



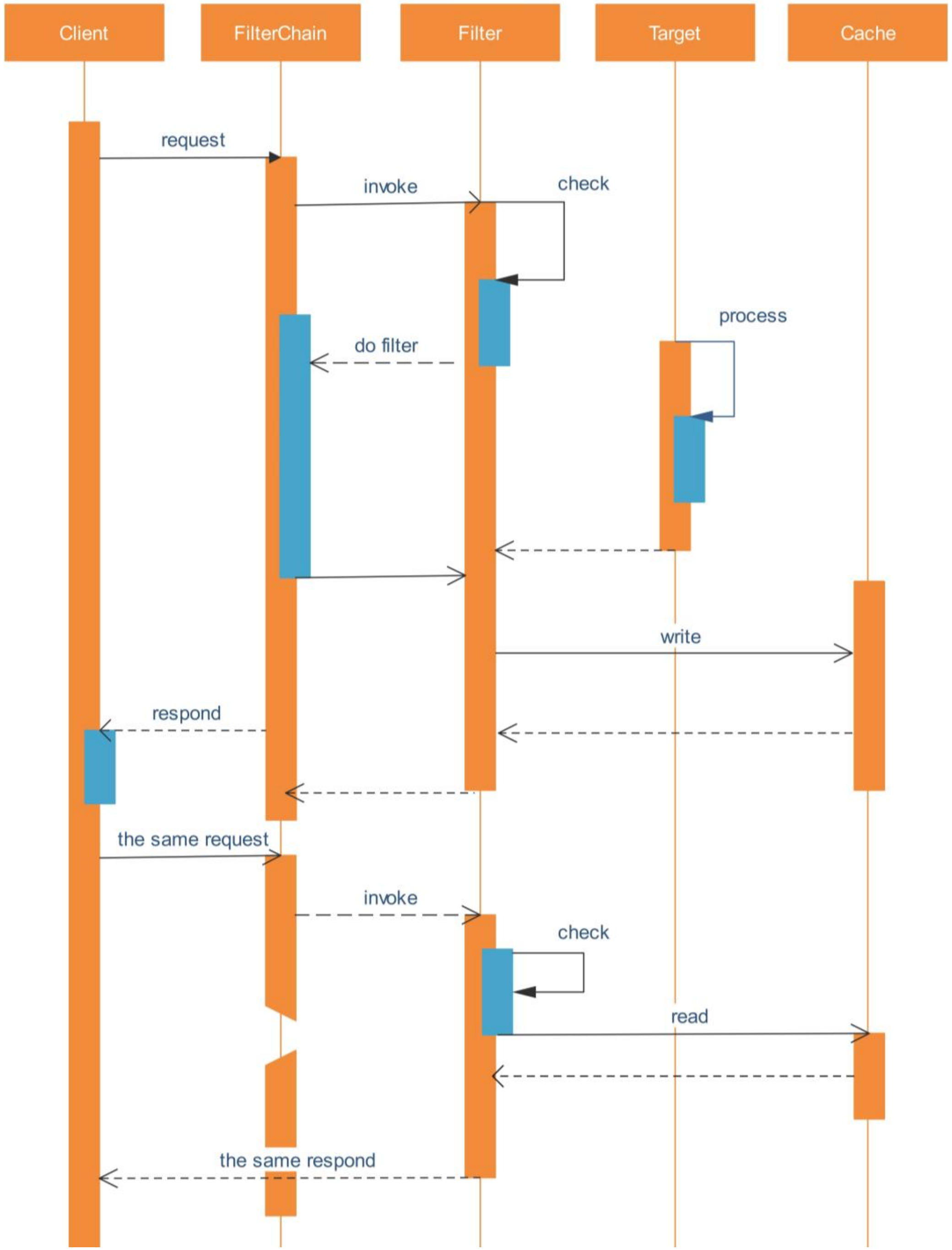
User flows

Models process and workflow



Sequence

Visualize how objects interact during method calls


Client
FilterChain
Filter
Target
Cache
request
invoke
check
process
do filter
write
respond
the same request
invoke
check
read
the same respond