

Energy

Adam Elí Davíðsson

Margrét Dís Stefánsdóttir

HBV501G

```
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class EnergyApplication {
8
9     public static void main(String[] args) { SpringApplication.run(EnergyApplication.class, args); }
```



About Us

Energy is an ecommerce website that sells caffeine related products.

For caffeine lovers who need to buy a variety of caffeine products without needing to go to the store.



Live Demonstration

Beginning:

- Review items.

- Search/filter items.

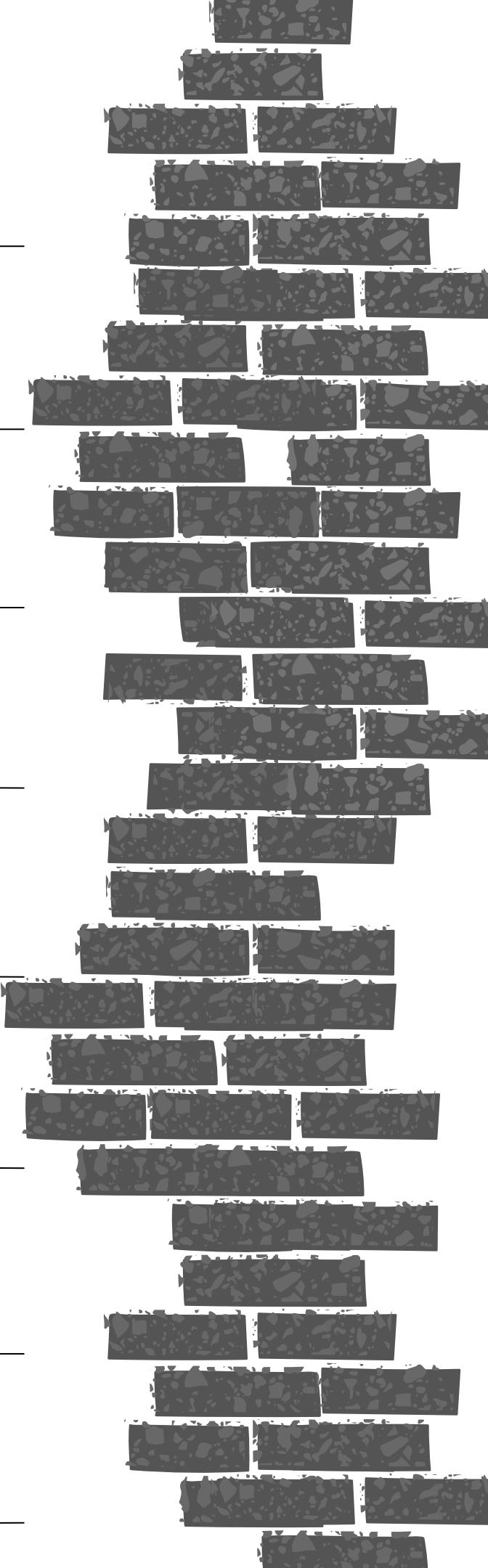
- Create an account.

- Caffeine calculator.

- Login.

- Process Sale.

- Write review.



Compromises

- Use cases "Write review" & "caffeine calculator" not priority.
- Added more use cases.

Key use cases

Admin

- Manage item
(Add/Remove/Edit).
- Manage order.
- Login.

User

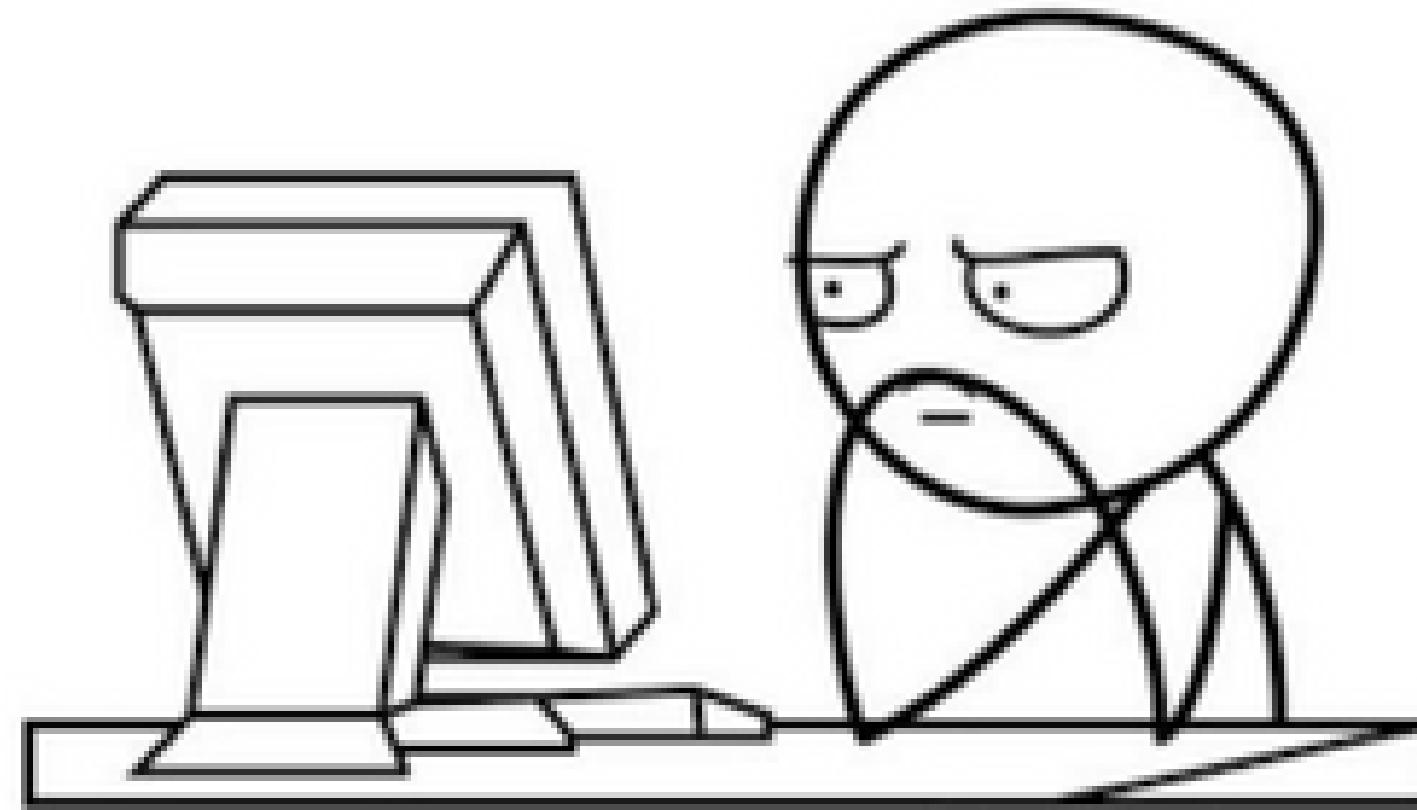
- Search/Browse items.
- Make purchase.
- Manage cart
(Add/Remove/Edit)
- Create/Edit account.
- Login & logout.



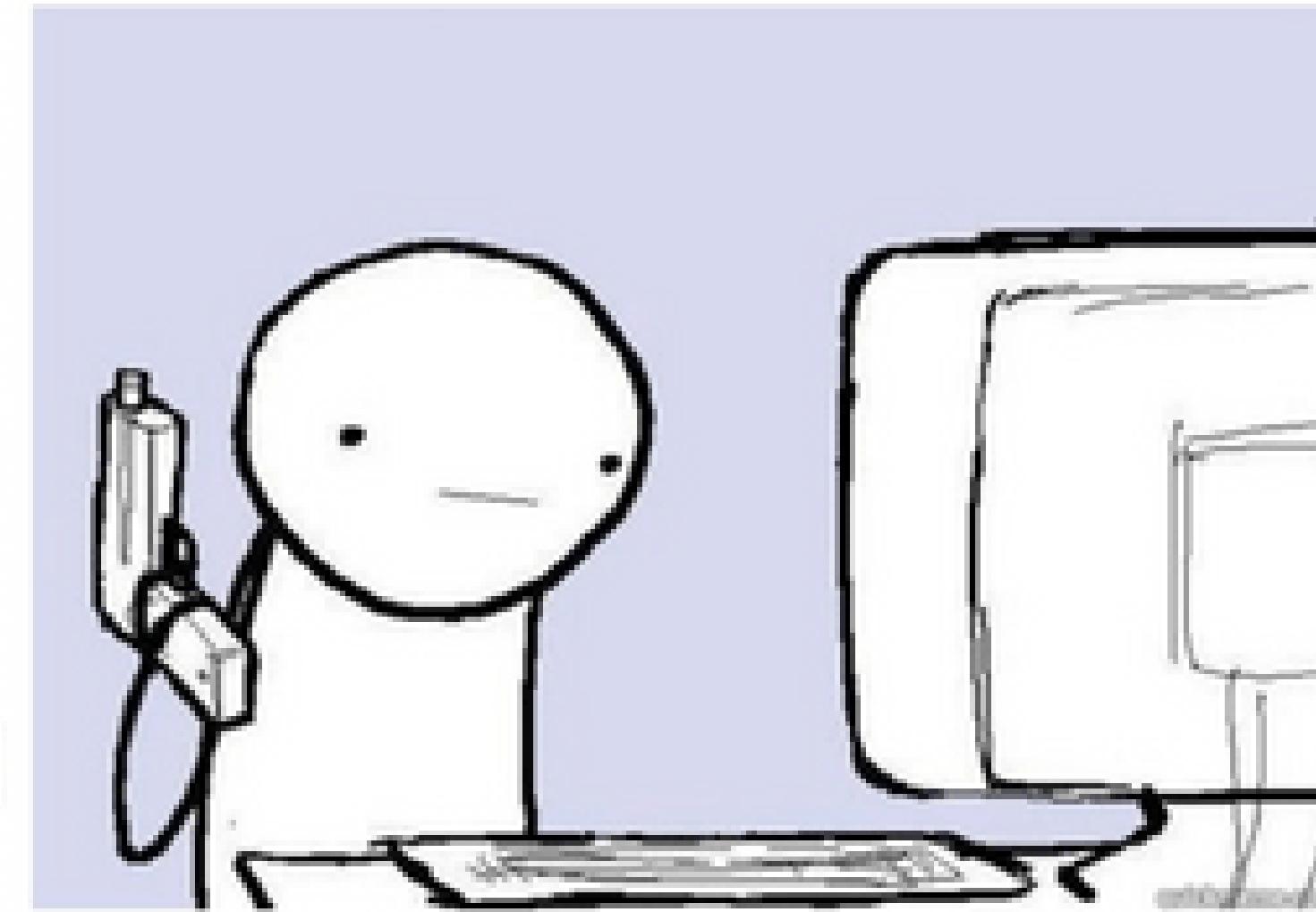


System architecture

Expectation

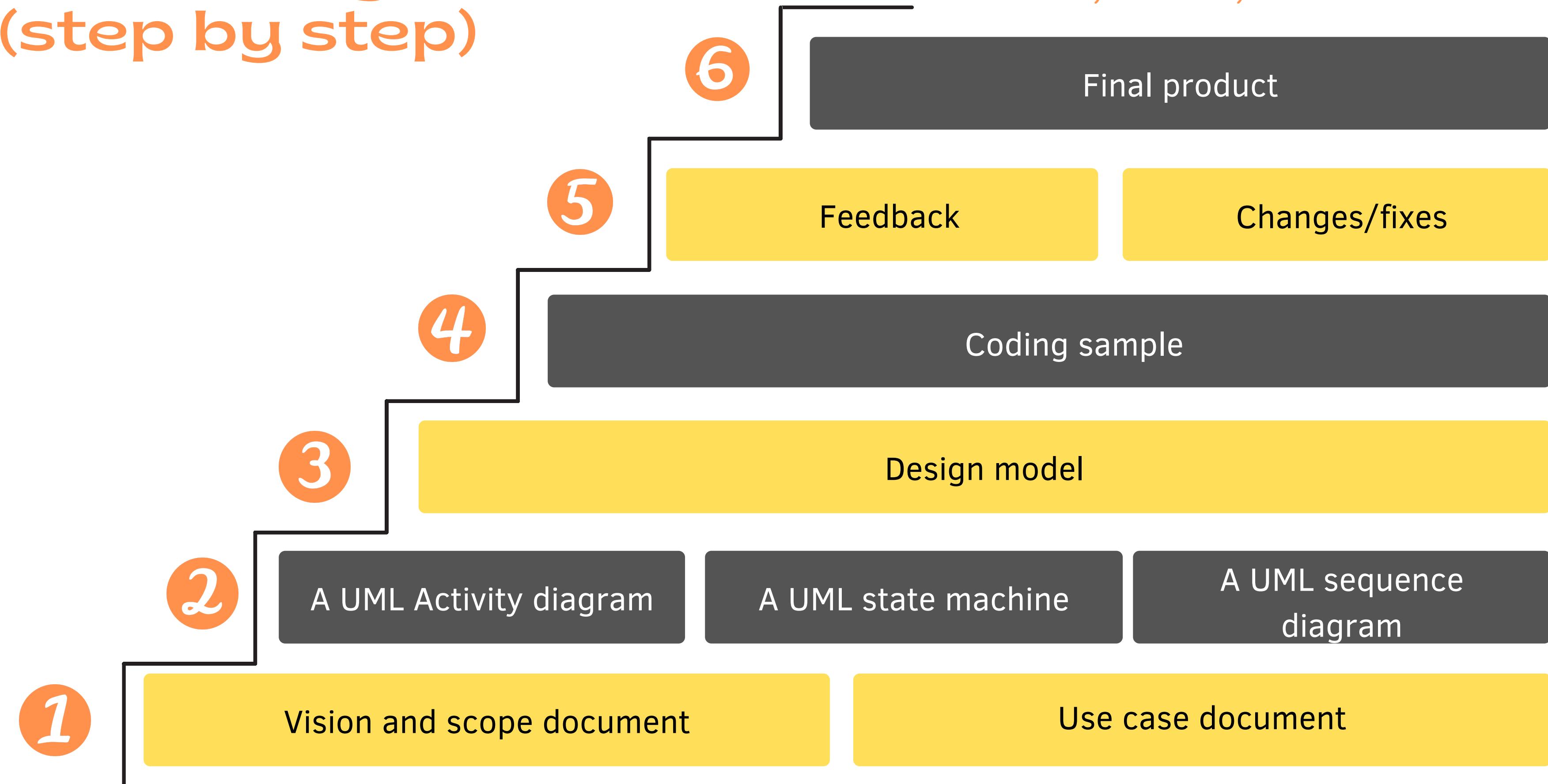


Reality



Our experiences

Planning (step by step)

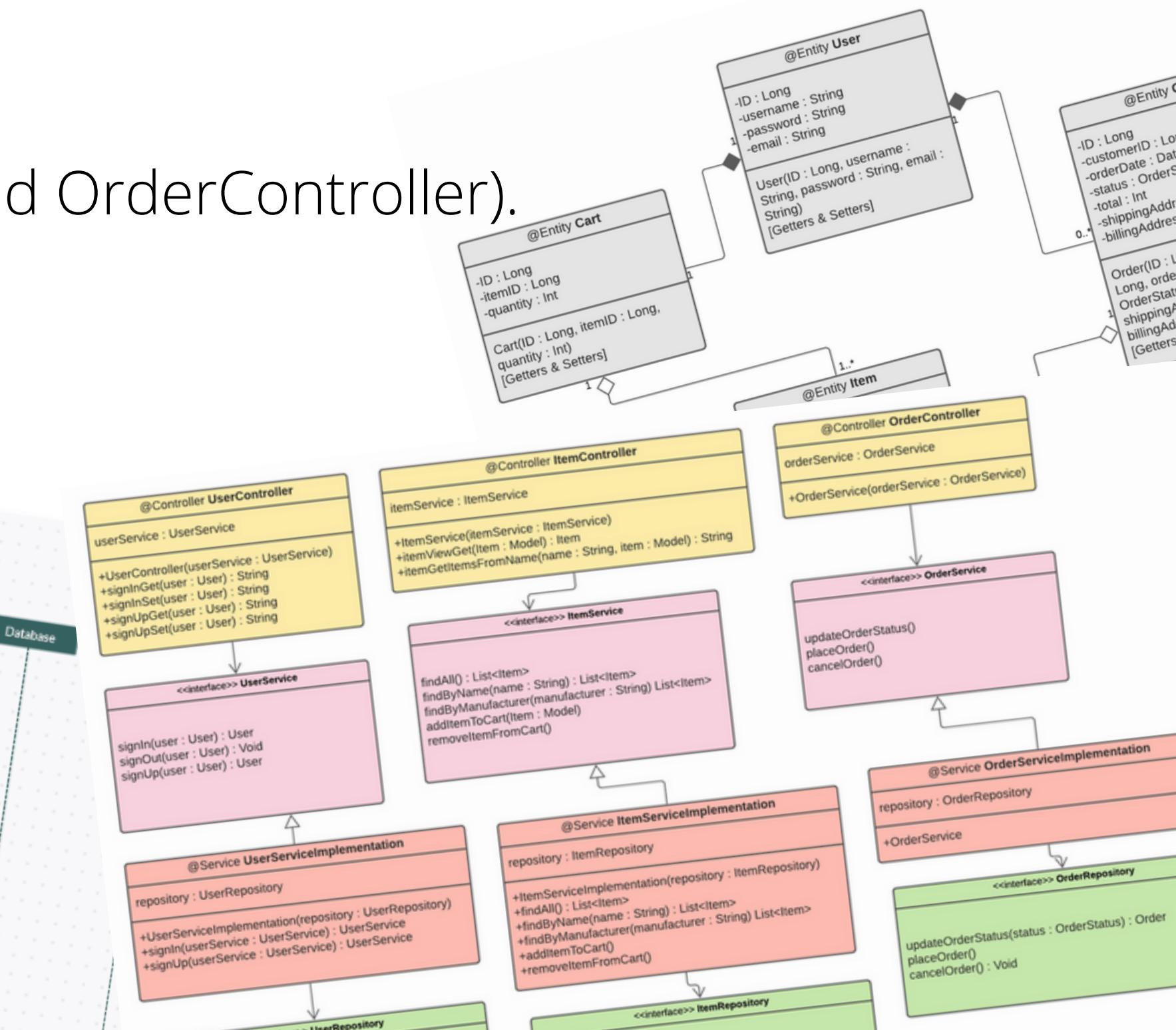
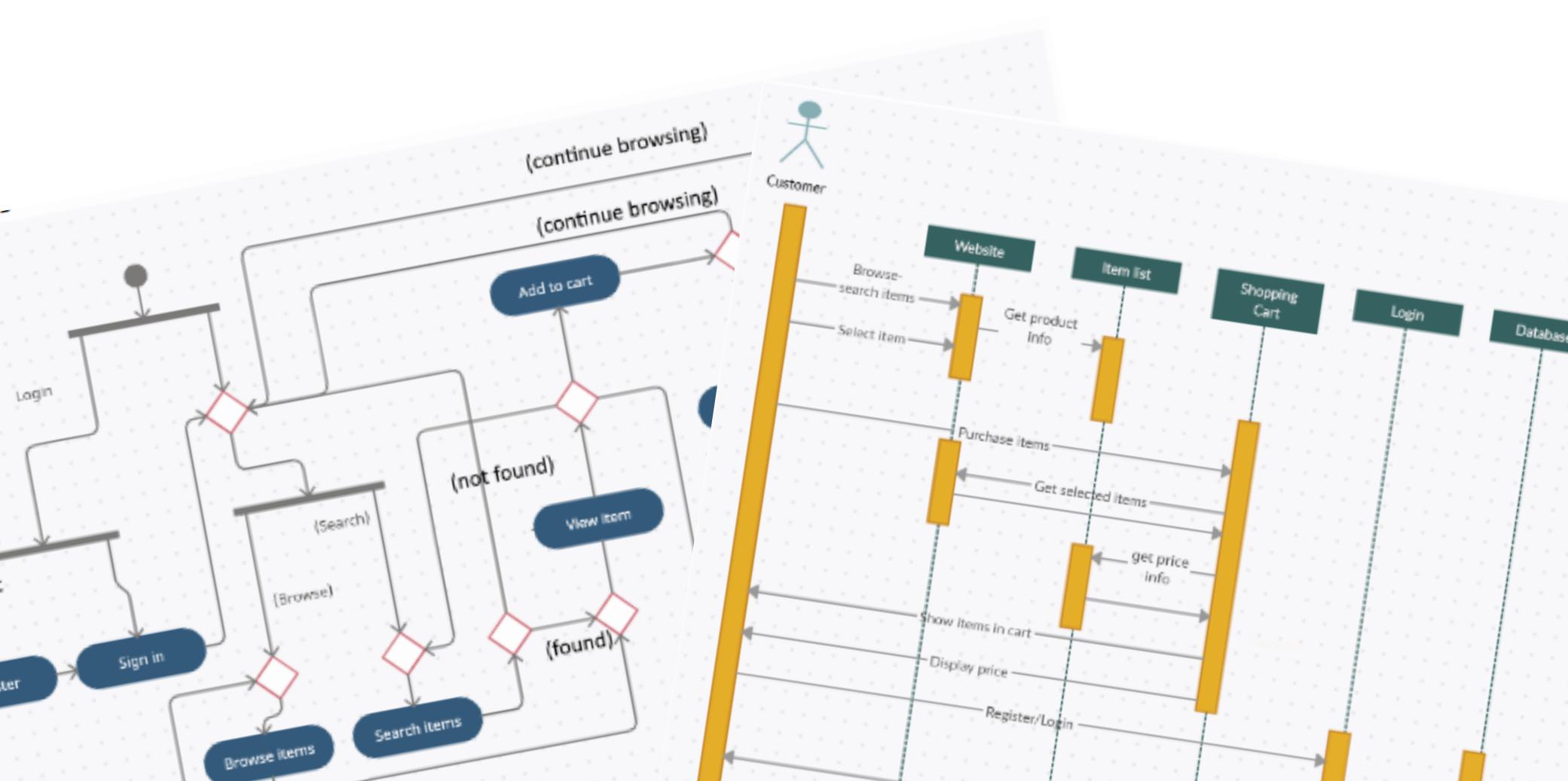


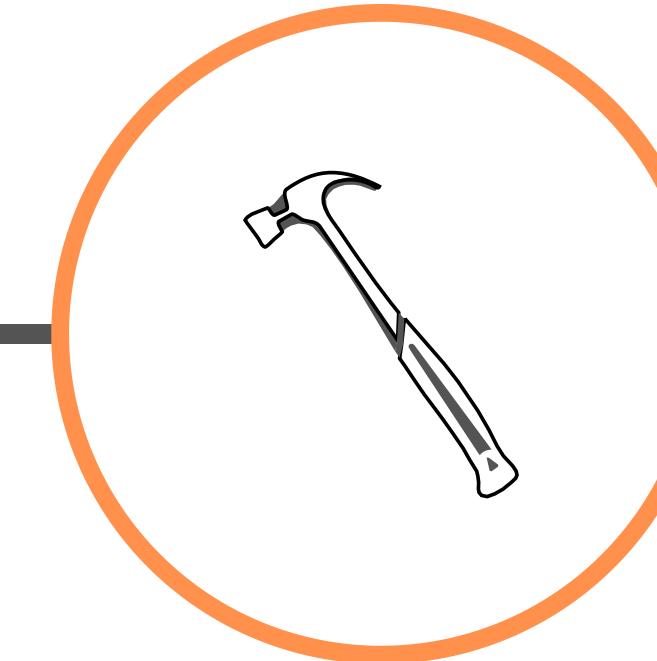
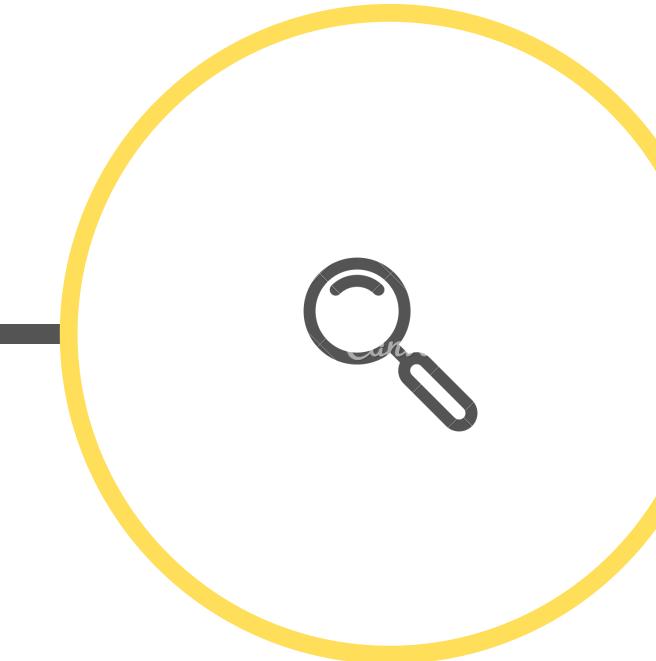
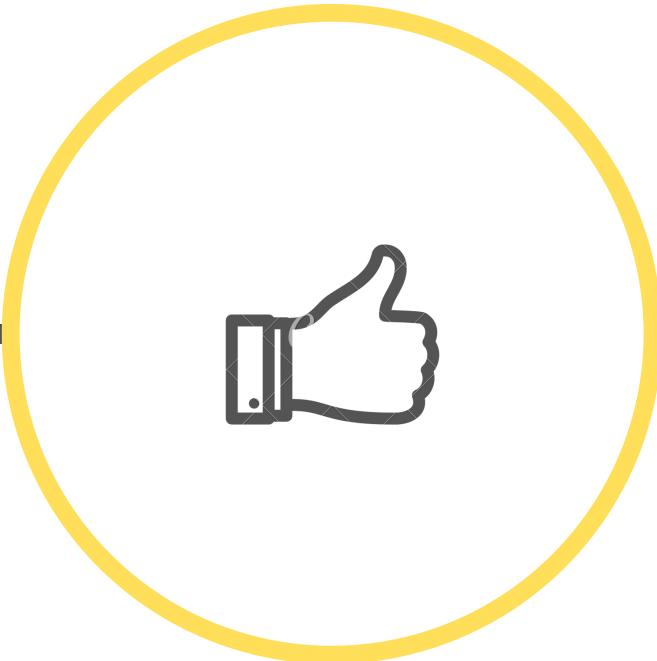
Overview of steps.

- Activity diagram & sequence diagram both include the process of ordering since our project is an e-commerce website.
- State diagram: login process.

Design model includes:

- 3 controllers (UserController, ItemController and OrderController).
- 4 entities (User, Item, Order and Cart)





WHAT WENT WELL?

- Use cases
- Communication

DIFFICULTIES:

- The state machine.
- The design model.
- Coding(didn't finish Order).

WHAT COULD WE DO DIFFERENT?

- Manage time better.
- Be better prepared.

FIXES:

- New design model
- Added use cases.
-

Recap: Aggregation and Composition

▪ Aggregation describes

- e.g. a Container holds Components
 - But is only weakly defined
 - and is just as well expressed
- Guideline: Don't bother

▪ Composition describes

- A part instance (e.g. Software Component)
 - Any part must belong to one composite
 - [Weaker semantics:
 - upon its creation,
 - upon its deletion,
 - Do not rely on people to remember "parts"
 - but make them explicit
- Guideline: Can be defined in a domain-specific language

THANK
YOU



for the technical design,
simplicity and "contains" label

ing that
ance (e.g. Board) at a time
gers")
on and deletion of parts, i.e.
parts
o another composite]
semantics,
al – if in doubt, leave it out



Questions?

