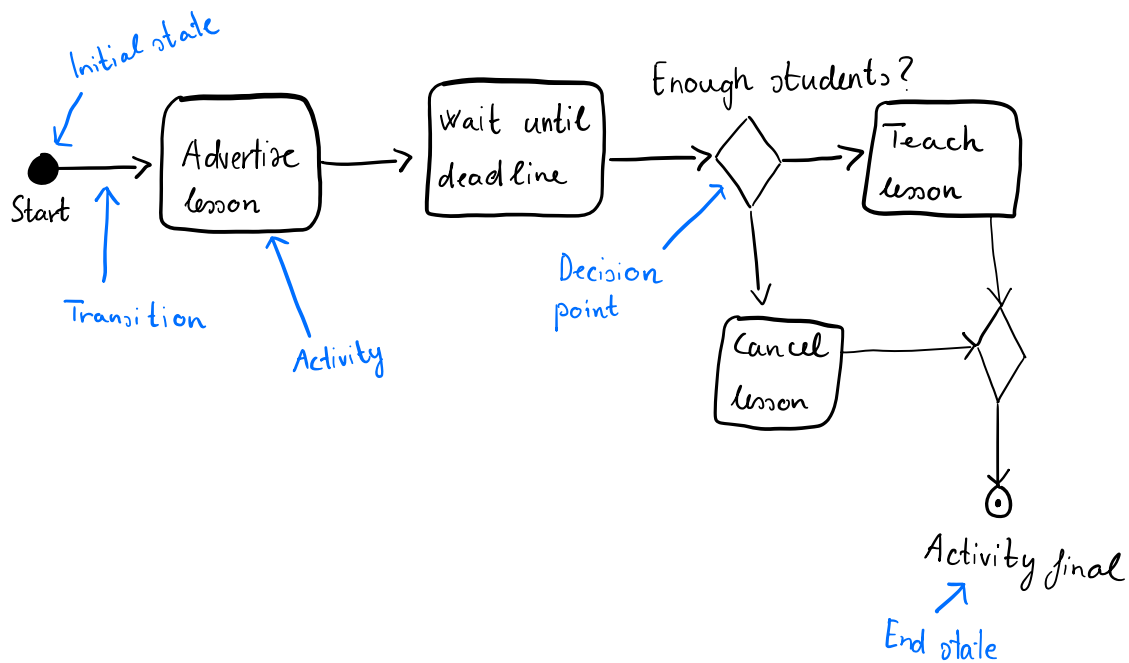


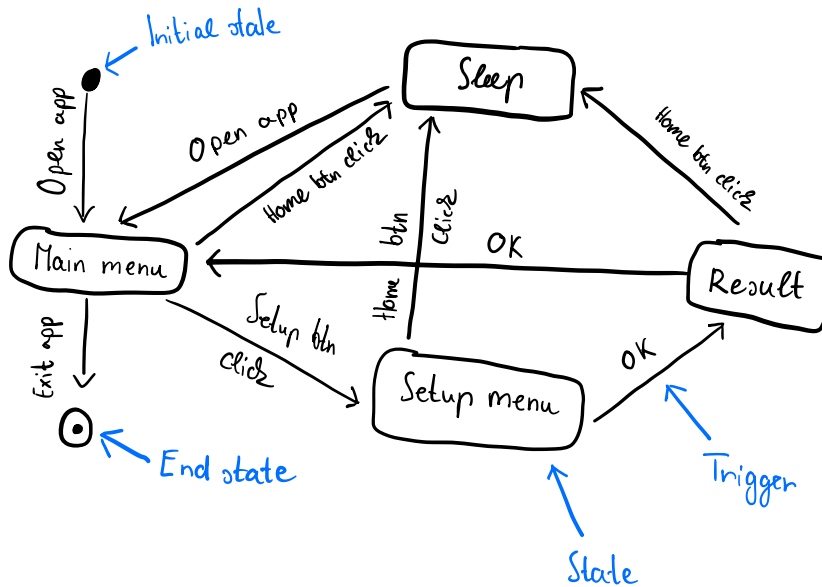
- Megrendelével való egyeztetés, hogy pontosan milyen szereplőket, funkciókat képzelt el, és ezeket a funkciókat melyik szereplővel lehet igénybe venni.
- Actor és használati eset közötti megfeleltetés.
- Van öröklődés:
  - Actorok között
  - Használati esetek között
- Tartalmazás/kibővítés → <<include>> / <<extend>>
  - Használati esetek között

# Activity diagram



- Use-case utáni következő lépés
- Rendszer folyamat leírása
  - Leírja az egyik tevékenységtől a másiké tartó sorrendet
  - Leírja a párhuzamos, elágazó és egyidejű folyamatát
- 1 használati eset = 1 Activity diagram
- Egy diagrammon belül más use-case-ek is előjöhethetnek
- Részei:
  - Initial state
  - End state
  - Transition
  - Activity
  - Decision point

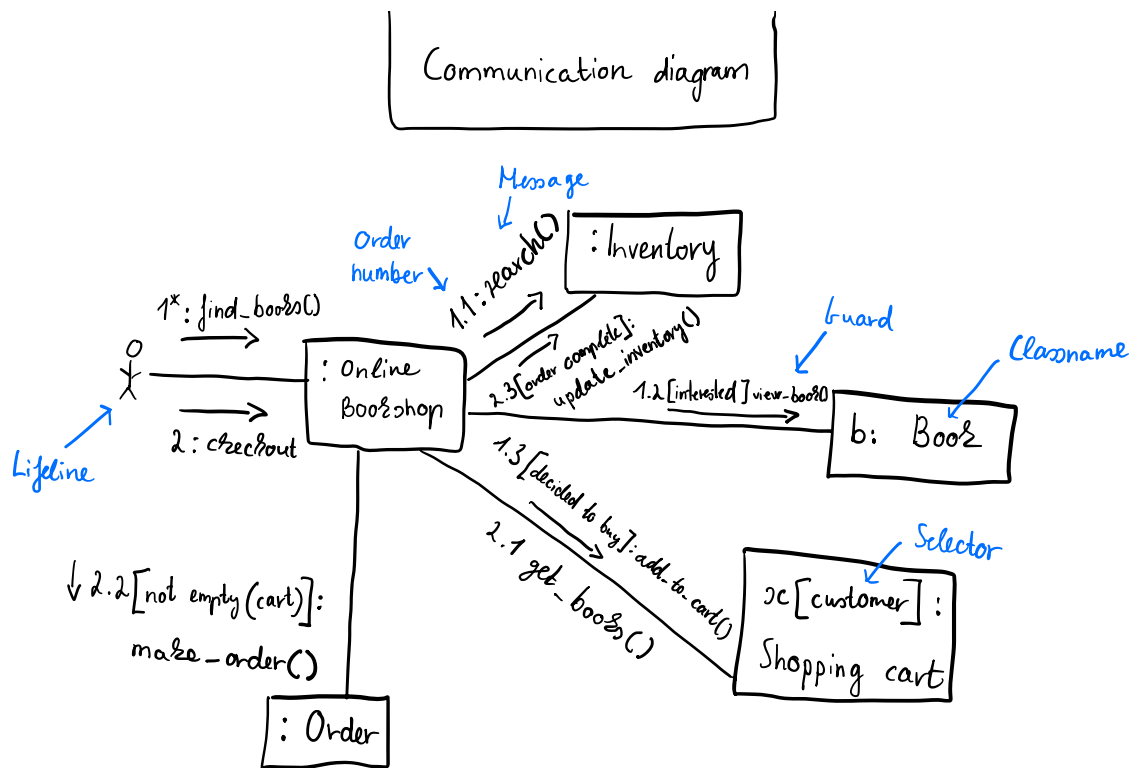
# State machine diagram



- Rendszer / objektum állapotainak egymás után következőségét ábrázolja.
- Irányított gráf, aminek csomópontjai a logikai állapotok, amikéi a közöttük lévő átmenetek.
- A végrehajtható műveletek az állapotokhoz és az átmenetekhez is tartoznak.
- Probléma, hogy egy idő után átláthatatlan lesz a diagram.
  - State transition
- Részei:
  - Initial state
  - End state
  - State
  - Trigger

## Interaction diagram

- Egy folyamat résztvevői (aktőrök vagy modulok vagy rétegek) közötti kommunikációs folyamatokat ábrázoljuk.
- 3 különböző diagramot különböztet meg:
  - Communication diagram: résztvevők és portok
  - Timing diagram: időzítési információk, megkezdések és állapotok is
  - Sequence diagram: ciklusok, feltételek és elágazások is.
- Mindhárom típus ugyanarra a célra való, csak más mélységben mutatja be a kommunikációs folyamatot.



- Behavioral diagram

- Rései:

- Lifeline: Felhasználó hogyan tud az osztályokkal kommunikációba kerülni.

- Message: Vmilyen metódus végrehajtása

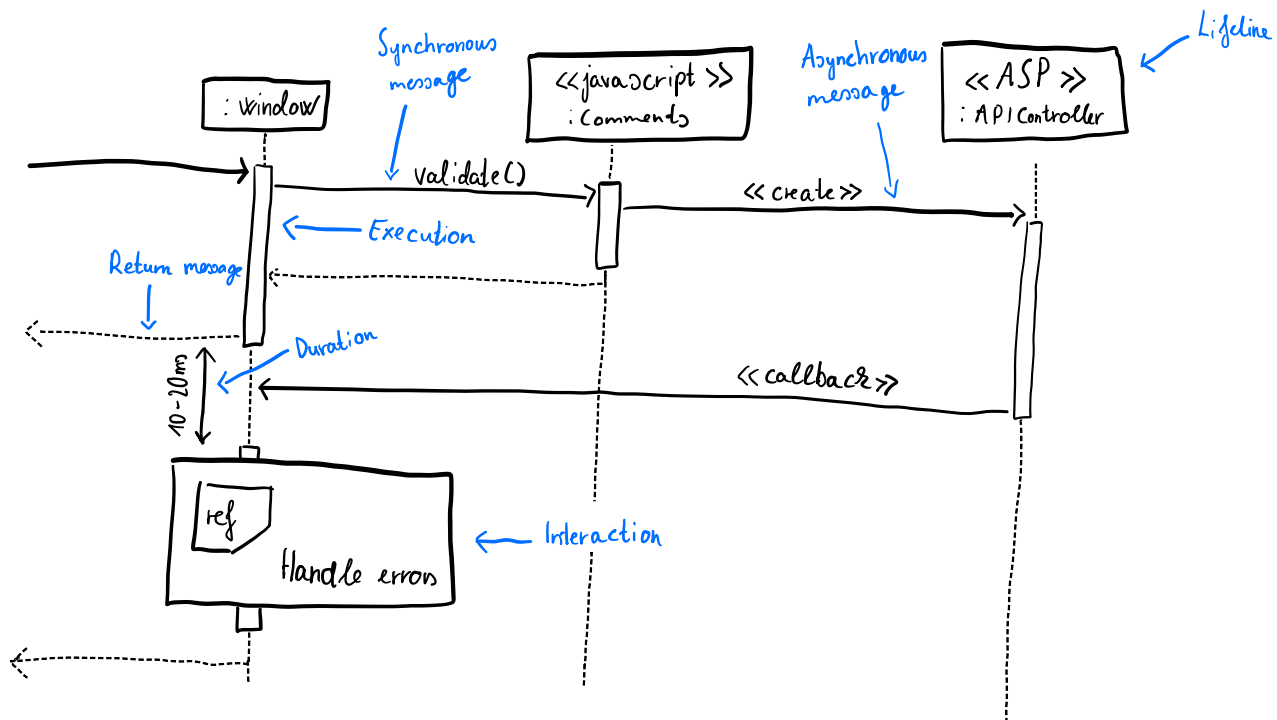
- Order number: Sorrendet írja le.

- Guard: „Őrszem”, csak akkor hajlandó végére ha xy

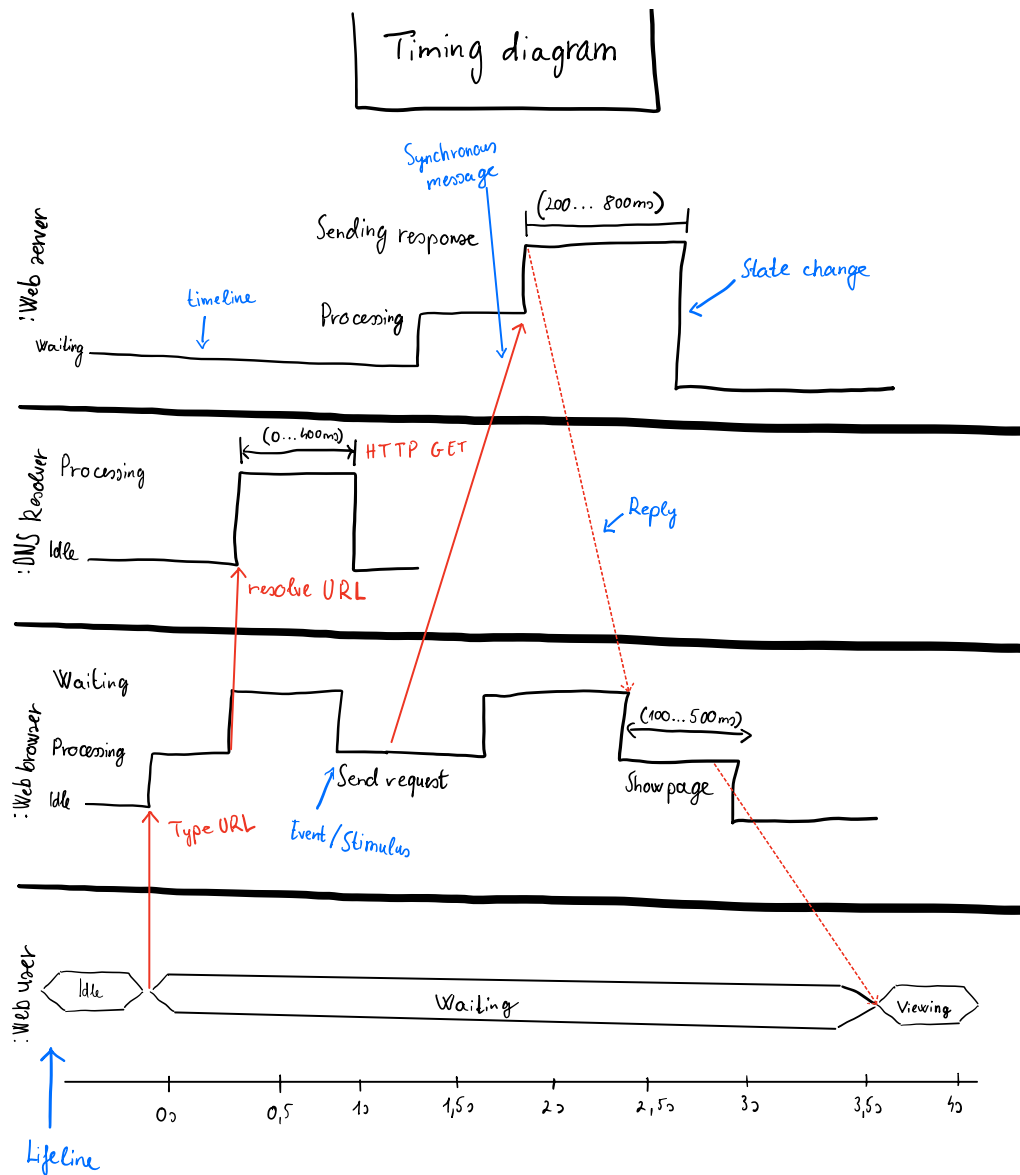
- Classname: Osztály neve

- Selector: Paramétereket adhatunk át, jelzésre jó.

# Sequence diagram



- Validáció, majd azután válasz, de validáció közben végrehajtható egy másik folyamat.
- Lifeline: Pl egy ApiController
- Synchronous message: Visszaad üzenetet
- Asynchronous message: Nem ad vissza üzenetet
- Return message: Vissza ad egy értéket vagy üzenetet



- Részai:

- Lifeline: "Életvonal"

- Timeline: "Idővonal", ami a lifeline-on belül mutatja, hogy az "timeline"-ja az a "különböző" állapotok között hogyan változik.

- Synchronous message: Kérést küldünk az egyik lifeline-ből a másikba.

↳ Időzítés miatt nincs Asynchronous Message-ek.

- Reply: válasz

- State change: Függetlenes vonalat az állapot változások.

- Event/stimulus: "Címke", ami valamilyen trigger jelöl.