System Development 2

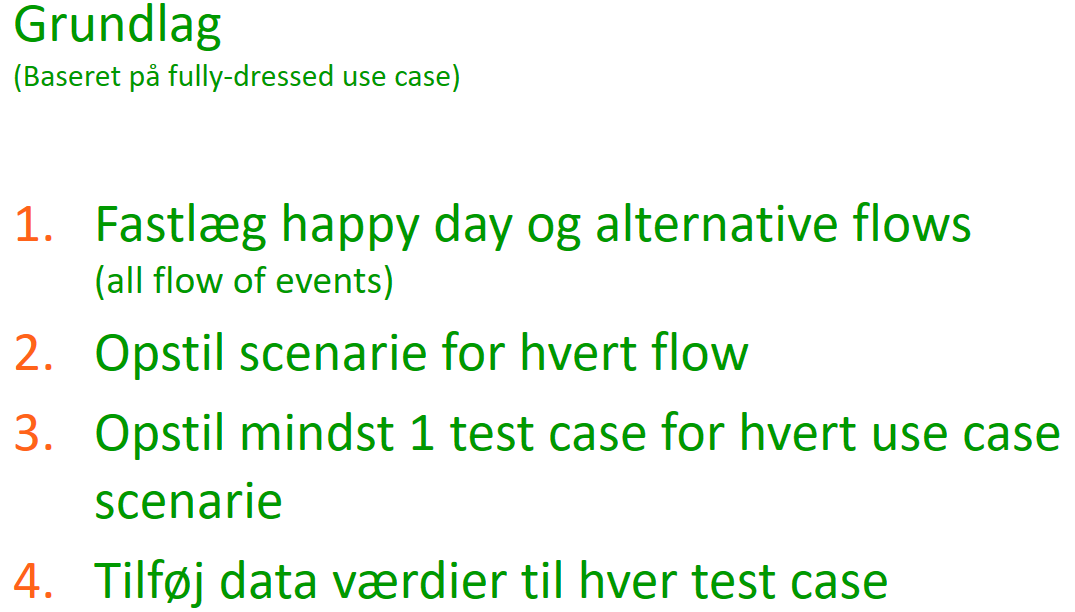
**Exercise: Test cases for system test**

**Lav test cases ud fra fully-dressed use case beskrivelse**

Udvælg en fully-dressed use case beskrivelse fra 1. semester projektet, alternativt brug use casen på næste side.

Udvid om nødvendigt beskrivelsen med flere og mere præcise alternative flows.  
Outcome of each alternative flow must be clear!

*Brug fremgangsmåden til at lave test cases:*



*De opstillede cases er styrende for JUnit tests!*

|  |  |  |
| --- | --- | --- |
| Use Case Name | **Withdraw money** | |
| Primary Actor | Customer | |
| Preconditions | Within hours – ATM open | |
| Postconditions | Money handed out and the transaction completed | |
| Frequency of Occurance | 15 times per hour. Peak ours occur. | |
| Flow of events | **Actor** | **System** |
| 1. Customer approaches to withdraw some money |  |
| 1. Customer indentifies an account | 1. Finds account informations and asks for authentification |
| 1. Indicates authentication info | 1. Validates info and presents ok |
| 1. Specifies amount | 1. Checks amount within withdrawel boundary and less or equal to balance. Pays out money and registers the transaction. When ok enters state: Ready |
| Alternate flow | 2a: Account not found  1: System informs user  4a: Authentification info wrong.   1. System informs customer .. 2. Systems asks if customer wants to try again or cancel   2a. Customer cancels  2b. Customer wants to try again. Customer repeats from step 4 until number of tries greater/equal 3  7a: Amount greater than withdrawel boundary.   1. System informs customer. 2. Systems asks if customer wants to specify another amount or cancel   2a. Customer cancels 2b. Customer wants to specify another amount Customer repeats from step 6  7b: Amount > balance  7c. Pay out money failure | |