Account lifecycle

As	a person accountable for privacy (data protection)
I want	user accounts that have a well-defined lifecycle with all the necessary account states
So that	we can manage users' personal data properly.
Description	User accounts need to have a well-defined lifecycle. For example, accounts at ChocAn need to be created; old accounts' data may need to be purged (personal data handling requirements); accounts may need to be flagged or locked because of unpaid account fees; new accounts may need to go through specific validation steps to be fully enabled (for example, require valid funds and creating a new and valid account number).
Acceptance criteria	» A state machine description of account states- validated, invalid number, or member suspended- is documented for maintenance. » » » Test cases exist that take an account through all the possible states of an account according to the state machine description. » Negative test cases exist that try out activities in various states of an account that should not be possible in those states, such as obtaining service with an invalid account, and verify that the activities fail
Refinement questions	» How and by whom are user accounts created? » How and when are user accounts destroyed? » What sort of "special states" can user accounts be in? » Have you thought about failing user interactions (e.g., registration failures) and in which state they will leave the user account? (i.e., not able to obtain service from service providers even though info is valid?)

Availability

As	a user
I want	the ChocAn services to be available when needed
so that	I get the expected value out of it.
Description	Availability is about the ChocAn services being available for use when required. The application development needs to consider both random faults and intentional attacks. A reasonable effort needs to be made to ensure availability in both cases. Fault sources include, for example, hardware failures, power and communication outages, unavailable service providers, and bugs. Intentional attacks typically target capacity (network, memory, storage, CPU time) or logical flaws (deadlocks, livelocks). Intentional attacks actively try to exploit worst-case scenarios.
Acceptance criteria	» Tests exist that introduce synthetic failures that simulate availability problems. » The service providers must be able to show when and when they are not available. » Customers must be able to book appropriate and available times for service
Refinement questions	» What are our availability requirements? » What types of failures could cause availability problems? » Does our architecture have points that are especially susceptible to denial-of-service attacks?