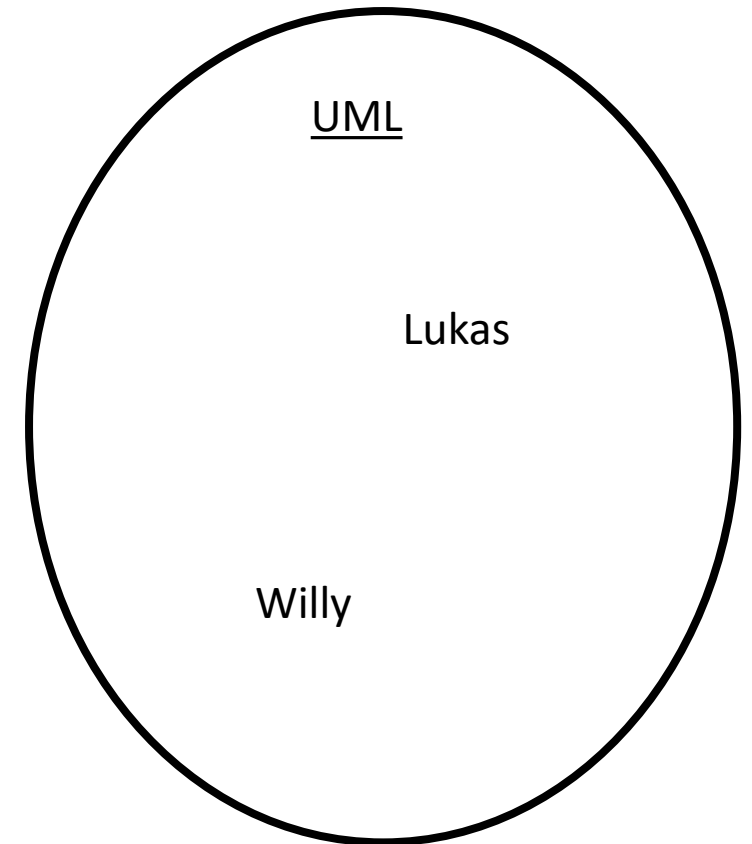
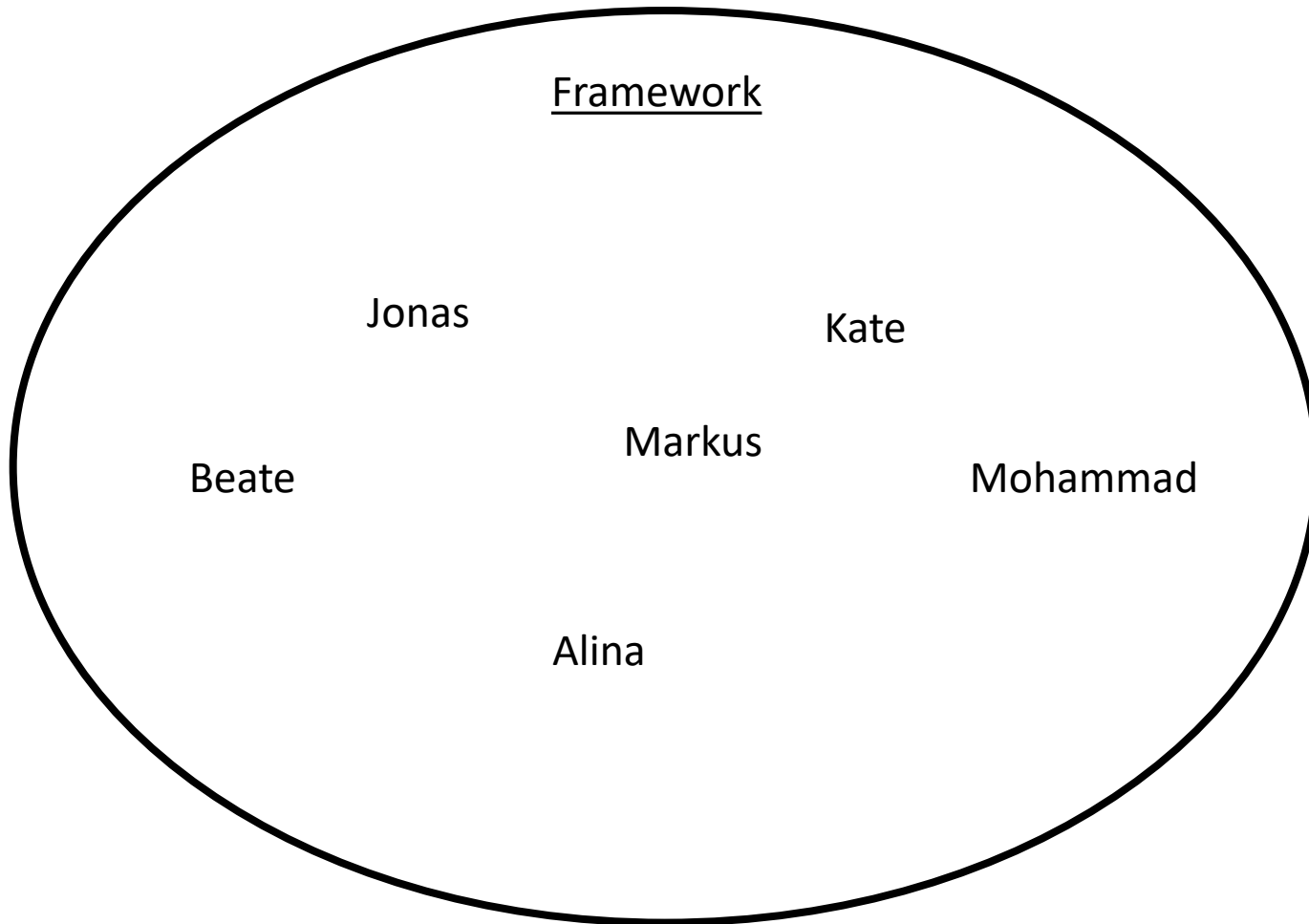


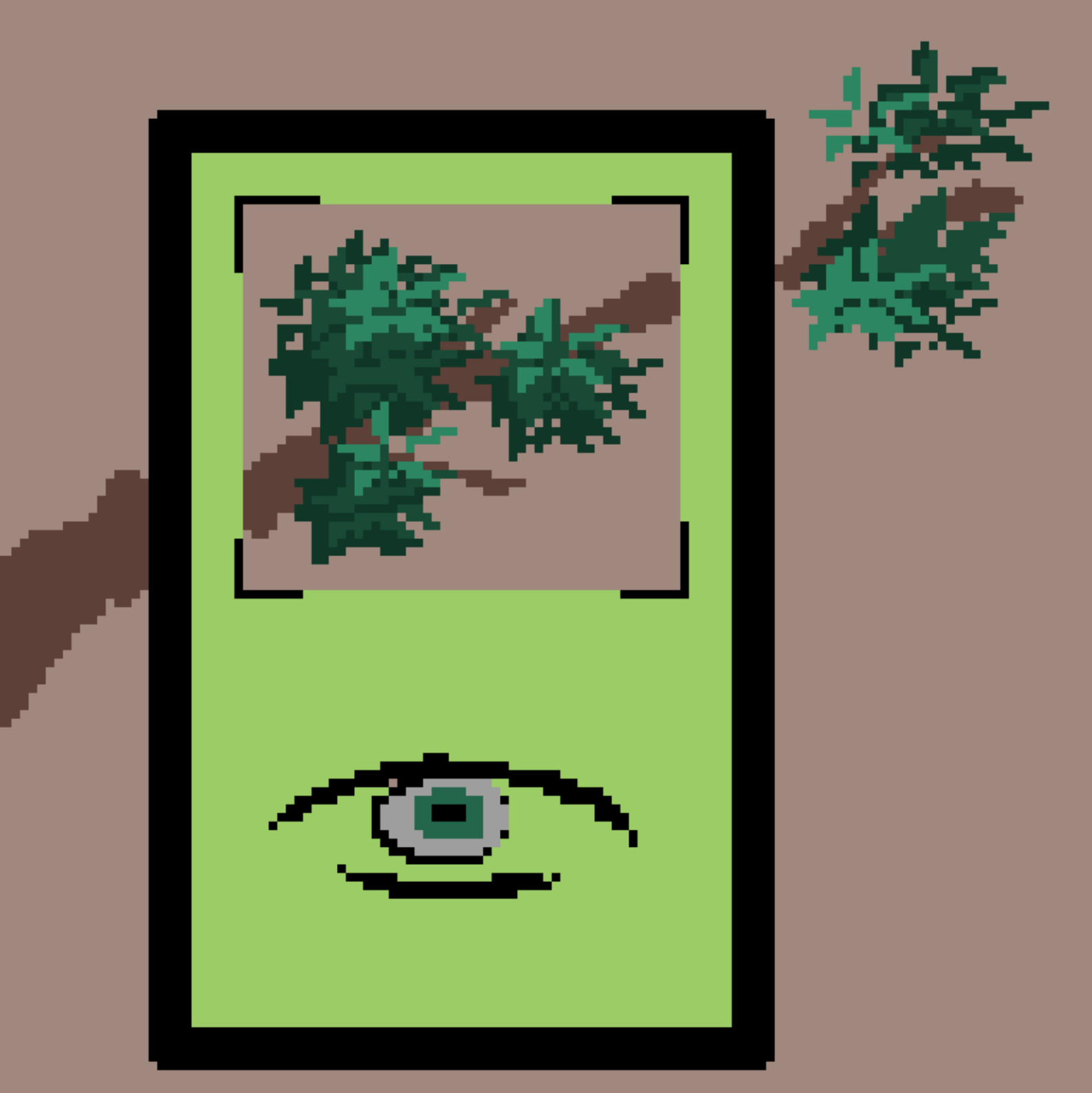
Meilenstein: UML

Gliederung

- Kleine Einführung (Willy)
- UML im Detail erklärt (Lukas)
- Arbeit der Framework-Gruppe (Jonas)

Einführung – Wie haben wir die letzten zwei Wochen gearbeitet?





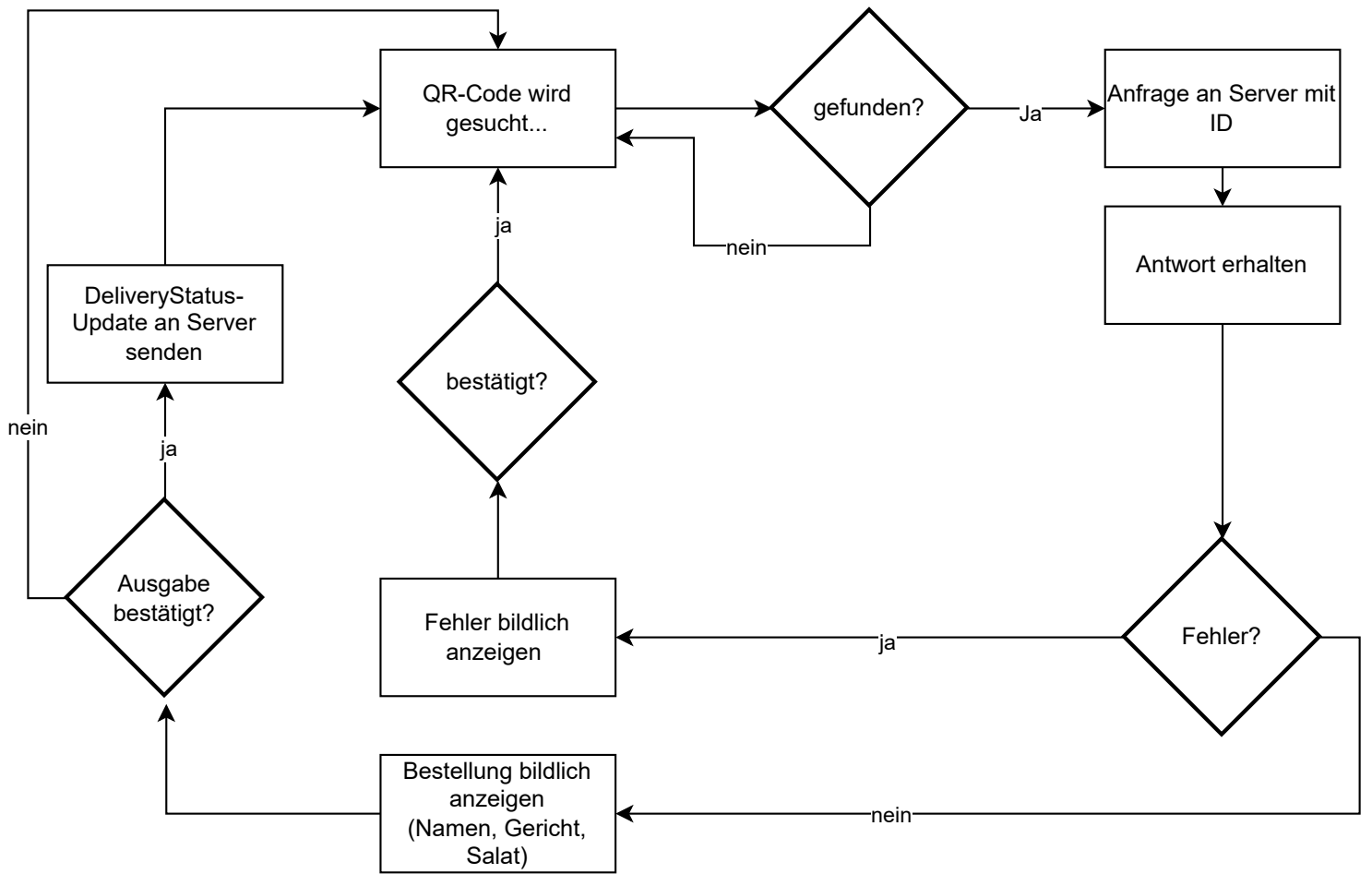
NAME



SALAD



Ausgabe-Flow



ID

PASSWORD

OK

U+

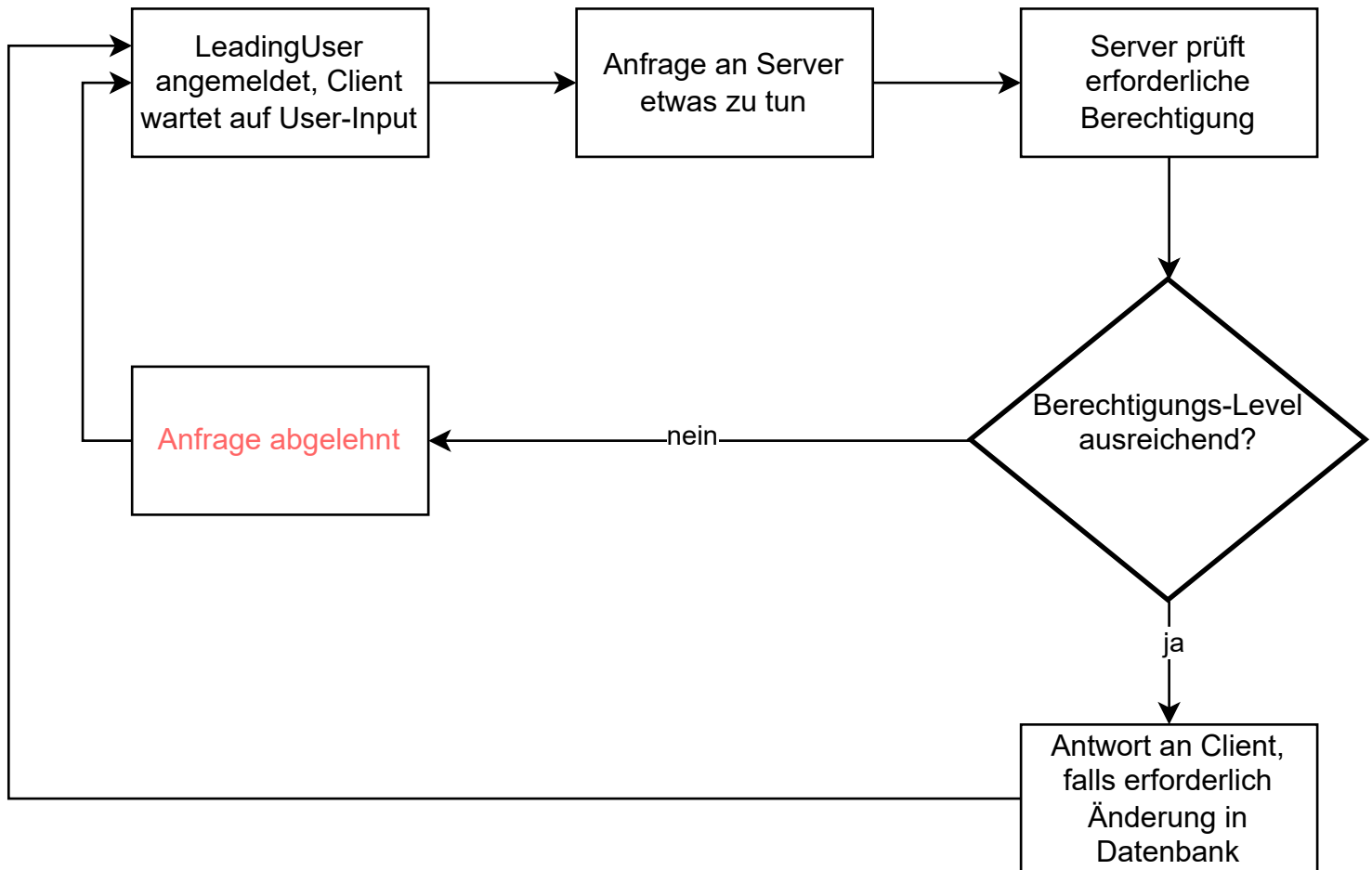
U-



U+

U-

LeadingUser-Flow



"Aktive Klassen"

Diese Klassen benutzen die passiven Klassen um ihre Funktion zu erbringen

Der KitchenClient kann nur einen QR-Code scannen und die ID an den Server schicken, sowie die Antwort grafisch darstellen

Der Server Manager nimmt alle persistenten Änderungen in der Datenbank vor

KitchenClient
- isSearching : bool - foundID : int
+ searchQrCode() : int + sendIDToServer() : void + showServerAnswer() : string + updateServerDeliveryStatus() : void

ServerManager
+ field: Type // habe ich hier gelassen um zu verdeutlichen, dass diese Klasse // keinen Status besitzt, so gesehen ist sie funktional
+ getUserFromId(int id): User + getTodaysOrderFromUser(User user): Order + sendOrderToKitchenClient(Order order): void + sendErrorToKitchenClient(Order order): void + updateDeliveryStatus(User user, int newStatus): void + changeOrderForUser(User user, orderId int, newOrder Order) + DeleteOrderForUser(User user, orderId int) + PlaceNewOrderForUser(User user, Order newOrder) + createNewOrder(string date, string mealName, bool salat, User owner) + registerUser(string name) + changeUserId(int oldId, int newId) + deleteUser(int id) + registerNewLeadingUser(int privilegeLevel, string name, int-array groupMembers) + addUserToGroup(LeadingUser groupChairman, int userID) + createHierarchyView(int id)

Der "LeadingClient" ist für alle Gruppen identisch, aber je nachdem welche Sicherheitsfreigabe ein Account hat, wird der client verschiedene Funktionen anbieten.

Beispielsweise wird die Gruppenleitung die Schaltfläche, die einen neuen LeadingUser registriert, nicht einmal angezeigt bekommen, da ihr "privilegeLevel "

zu gering ist. Darüber hinaus wird aber der ServerManager jede Anfrage darauf kontrollieren, ob der Anfragende die notwendige berechtigung hat,

da ansonsten eine Manipulation sehr einfach wäre (Die Stelle im Arbeitsspeicher finden, wo das privilegeLevel gespeichert ist, und diese mit etwas anderem überschreiben)

LeadingClient
- loggedIn: bool - whoAmI: LeadingUser - privilegeLevel : int
+ registerNewLeadingUser(string Name, int privilegeLevel, string password, int-array groupmembers): void + logMeIn(int id, string password) : string // sollte der account in der Datenbank gefunden werden, steht in //diesem String codiert, wie hoch die Berechtigung ist und //informationen wie der Name + registerNewUser(string Name, LeadingUser teamleader) : string + readUserFromCSV(string path) : void + showLocalGroupsAndOrders() : void + showWorkshopGroups() : void + setSubstituteLeader(LeadingUser sickLeadingUser, LeadingUser substituteLeader): void + placeOrderForUser(Order order, User user) : void + tryChangingOrderForUser(User user, Order newOrder) : void + showGroupmembersAndOrders() : void // braucht keine Inputs, LeadingUser hat referenzen zu //gruppenmitgliedern

"Passive Klassen"

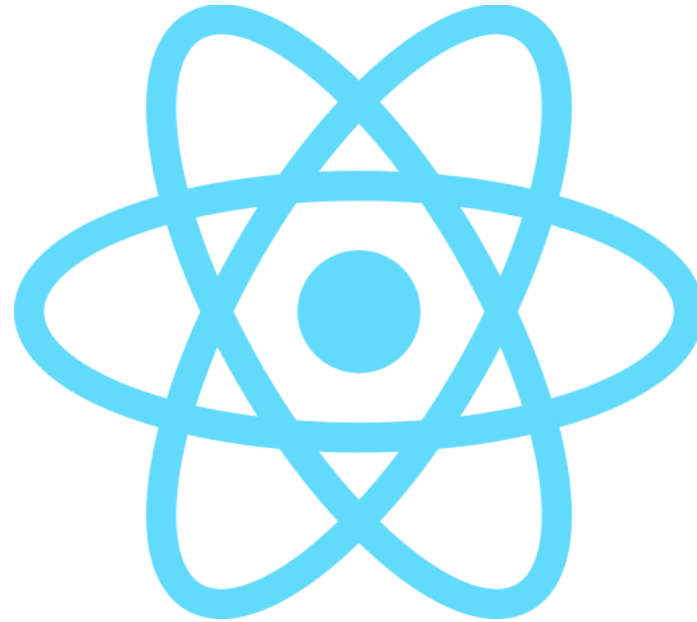
Objekte dieser Klassen tun selbst nichts - mit ihnen wird etwas getan. Sie dienen lediglich als Datenbehälter und werden von den aktiven Klassen genutzt.

User
- Id : int - Name : string
+ get...(): <T> + set...(): <T>

Leading User extends User
-PrivilegeLevel : int -Password : string -controlOver : int-array
+ get...(): <T> + set...(): <T>

Order
-date : string -orderId : int -mealName : string -salat: bool (-whoOrdered: User)
+ get...(): <T> + set...(): <T>

Auswahl der Frameworks



React – Frontend

Auswahl der Frameworks



Spring Boot – Backend

Auswahl der Frameworks



MySQL – Datenbank