UserService -userDAO : UserDAO +addUse RewardService shipDAO : ShipDAO
roomDAO : RoomDAO
crewDAO : CrewDAO
weaponDAO : WeaponDAO -userDAD : UserDAD : dearname : String) : void : +getUser(username : String) : User : +updateUser(u : User) : void : +fenoveUser(g\)\text{Vsername} : String) : void : +UserService(source : ConnectionSource) : void : +UserService(source : ConnectionSource) : void : +UserService(source : ConnectionSource) : void : +UserService(source : String) : +UserService(source : roomDAO : RoomDAO -shipDAO ShipDAO -weaponDAO: WeaponDAO
+weaponReward(s: Ship, dropTable: List<Weapon>): Weapon
+coipsReward(s: Ship, c: int): yold
+tpeiReward(s: Ship, f: int): yold
+cockerReward(s: Ship, r: int): yold
+crewReward(s: Ship, r; rew: List<Crew>): yold +validateSystemReplacement(s: String): boolean
+installGystem(ship: Ship, system: System, room: Room): void
+uninstalledSystem(ship: Ship, system: System): void RoomService
-shipDAQ© ShipDAQ
-roomBAQ : RoomDAQ
-tcapseBreach(ship: Ship, room? Room) : void
-tcapseBreach(ship: Ship, system : System) : void
-treEnable System(ship: Ship, system : System) : void
-treEnable System(ship: Ship, system : System) : void BattleService +giveCoins(s: Ship, amount, sint): void +removeCoins(s: Ship, amount: int): void +giveLoot(s: Ship, tropTable: Llst-{Wapon>, crewDro-meteor(shower(s: Ship): void +disableSystemsInNebula(s: Ship, duration: int): void wDropTable : List<Crew>) ; vo TravelService
shipDAO: ShipDAO
-planetDAO: PlanetDAO
+validateJump(s: Ship): boolean
+jump(s: Ship, dest: Planet): voice TraderSérvice

TraderDÃO: TraderDÃO

crewDÃO: CrewDÃO

vesaponDÃO: WeaponDÃO

weaponDÃO: WeaponDÃO

**ValidatePuchase(ship: Ship, price: intf: boolean

**purchase(Weapon(ship) Ship, trader, Trader, rewe, Orew): void

**purchase(Vew(ship) Ship, trader, Trader, rewe, Orew): void

**purchase(Vew(ship) Ship, trader, Trader, amount: int); void

**purchase(Vew(ship) Ship, trader: Trader, amount: int); void

**purchase(Vew(ship): Ship, trader: Trader, amount: int); void

**purchase(Vew(ship): Ship, trader: Trader, amount: int); void

**sellWeapon(ship): Ship, trader: Trader, amount: int); void

**sellWeapon(ship): Ship, trader: Trader, amount: int); void

**sellWeapon(ship): Ship, trader: Trader, amount: int); void CrewService -shipDAO : ShipDAO -crewDAO : CrewDAO -roomDAO : RoomDAO walidate(rewMove(s : String) : boolean

walidate(rewMove(s : String) : boolean

move(rewToRoom(ship : Ship, crew : Crew, room : Room) : void

healCrewMember(ship : Ship, srew : Crew, healAmount : int) : void

damage(rew(ship : Ship, sroom : Room) amount : int) : void

damage(rew(ship : Ship, sroom : Room, amount : int) : void

repairBreach(ship : Ship, room : Room) : void

repairBreach(ship : Ship, room : Room) : void HangarService +generateMap(username String, seed (int): Overworld +chooseShip(username String, shipDesign: String): Ship