# Object and Method Reference

## 1. Restaurant

stereotype: "«inherits SimEntity»"

constraint: "{manages operations}"

color: "#CCE5FF"

* "- name: String"
* "- tables: List<Table>"
* "- menu: List<Platter>"
* "- staff: List<Staff>"
* "- revenue: double"
* "- avgWalkTime: double"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ simulateDay()"
4. "+ decideStaffing()"
5. "+ decideTableAssignments()"
6. "+ decideInventoryRestock()"
7. "decideStaffing (uses Probability: RushHourOverloadChance)"
8. "decideInventoryRestock (uses Probability: IngredientSpoilageRate)"
9. "+ getGlobals(): GlobalVariables"
10. "+ getDecisionEngine(): GlobalMethods"
11. "+ evaluate(): boolean"
12. "+ commit(): void"
13. "+ getState(): String[][]"

## 2. Hostess

stereotype: "«inherits SimEntity»"

constraint: "{assigns tables}"

color: "#CCE5FF"

* "- name: String"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ greetCustomer(Customer)"
4. "+ seatCustomer(Customer, Table)"
5. "+ decideTableAssignment(Customer)"
6. "+ releaseTableDecision(Table, Customer)"
7. "+ decideSeating()"
8. "+ decideQueueOrder()"
9. "decideSeating (uses Probability: OptimalTableSelectionProbability)"
10. "decideQueueOrder (uses Probability: QueueBalkingProbability)"
11. "+ evaluate(): boolean"
12. "+ commit(): void"
13. "+ getState(): String[][]"

## 3. CashRegister

stereotype: "«inherits SimEntity»"

constraint: "{tracks revenue}"

color: "#CCE5FF"

* "- totalStoreIncome: double"
* "- totalTips: double"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ processPayment(Server, Order, double tip)"
4. "+ recordTip(Server, double tip)"
5. "+ decidePaymentHandling()"
6. "+ evaluate(): boolean"
7. "+ commit(): void"
8. "+ getState(): String[][]"

## 4. Table

stereotype: "«inherits SimEntity»"

constraint: "{seating location}"

color: "#CCE5FF"

* "- id: int"
* "- capacity: int"
* "- location: String"
* "- occupied: boolean"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ assignCustomer(Customer)"
4. "+ clearTable()"
5. "+ decideClearTable()"
6. "+ evaluate(): boolean"
7. "+ commit(): void"
8. "+ getState(): String[][]"

## 5. Platter

stereotype: "«inherits SimEntity»"

constraint: "{cookable item}"

color: "#CCE5FF"

* "- name: String"
* "- ingredients: List<Ingredient>"
* "- cookingTime: int"
* "- price: double"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ prepare()"
4. "+ evaluate(): boolean"
5. "+ commit(): void"
6. "+ getState(): String[][]"

## 6. Ingredient

stereotype: "«inherits SimEntity»"

constraint: "{perishable goods}"

color: "#CCE5FF"

* "- name: String"
* "- quantity: double"
* "- costPerUnit: double"
* "- expiryDate: Date"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ evaluate(): boolean"
4. "+ commit(): void"
5. "+ getState(): String[][]"

## 7. Cook

stereotype: "«inherits SimEntity»"

constraint: "{kitchen staff}"

color: "#CCE5FF"

* "- name: String"
* "- skillLevel: int"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ cook(Platter)"
4. "+ prep(Platter)"
5. "+ announceOrder(Table)"
6. "+ decideNextOrder()"
7. "+ decideCookingPriority()"
8. "decideNextOrder (uses Probability: PrepTimeVariance)"
9. "decideCookingPriority (uses Probability: CookingDelayRate)"
10. "+ evaluate(): boolean"
11. "+ commit(): void"
12. "+ getState(): String[][]"

## 8. Server

stereotype: "«inherits SimEntity»"

constraint: "{floor staff}"

color: "#CCE5FF"

* "- name: String"
* "- tablesAssigned: List<Table>"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ serve(Table)"
4. "+ takeOrder(Customer)"
5. "+ deliverCheck(Customer)"
6. "+ returnPayment(CashRegister)"
7. "+ decideDeliverySequence()"
8. "+ decideTipHandling()"
9. "decideDeliverySequence (uses Probability: ServiceDelayVariance)"
10. "decideTipHandling (uses Probability: TipLikelihood)"
11. "+ evaluate(): boolean"
12. "+ commit(): void"
13. "+ getState(): String[][]"

## 9. ServerTable

stereotype: "«inherits SimEntity»"

constraint: "{pickup counter}"

color: "#CCE5FF"

* "- bell: boolean"
* "- orderQueue: Queue<Order>"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ announceOrder(Order)"
4. "+ pickUpOrder(Server)"
5. "+ viewNextOrder()"
6. "+ evaluate(): boolean"
7. "+ commit(): void"
8. "+ getState(): String[][]"

## 10. Order

stereotype: "«inherits SimEntity»"

constraint: "{food request}"

color: "#CCE5FF"

* "- platter: Platter"
* "- table: Table"
* "- timePlaced: int"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ getTable()"
4. "+ getPlatter()"
5. "+ calculateTip(happiness: double)"
6. "+ evaluate(): boolean"
7. "+ commit(): void"
8. "+ getState(): String[][]"

## 11. Customer

stereotype: "«inherits SimEntity»"

constraint: "{dining patron}"

color: "#CCE5FF"

* "- groupSize: int"
* "- patience: int"
* "- satisfaction: int"
* "- tablePreference: String"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ requestTable(String)"
4. "+ order(Platter)"
5. "+ leave()"
6. "+ decideExit()"
7. "+ decideReorder()"
8. "decideExit (uses Probability: PatienceThreshold)"
9. "decideReorder (uses Probability: ReorderLikelihood)"
10. "+ evaluate(): boolean"
11. "+ commit(): void"
12. "+ getState(): String[][]"

## 12. Inventory

stereotype: "«inherits SimEntity»"

constraint: "{stock control}"

color: "#CCE5FF"

* "- ingredients: List<Ingredient>"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ restock()"
4. "+ checkStock(Platter)"
5. "+ decideRestockTiming()"
6. "+ evaluate(): boolean"
7. "+ commit(): void"
8. "+ getState(): String[][]"

## 13. Refrigerator

stereotype: "«inherits SimEntity»"

constraint: "{cold storage}"

color: "#CCE5FF"

* "- capacity: double"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ store(Ingredient)"
4. "+ remove(Ingredient)"
5. "+ evaluate(): boolean"
6. "+ commit(): void"
7. "+ getState(): String[][]"

## 14. Stove

stereotype: "«inherits SimEntity»"

constraint: "{hot line}"

color: "#CCE5FF"

* "- slots: int"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ startCooking(Platter)"
4. "+ finishCooking(Platter)"
5. "+ evaluate(): boolean"
6. "+ commit(): void"
7. "+ getState(): String[][]"

## 15. Finance

stereotype: "«inherits SimEntity»"

constraint: "{tracks profit}"

color: "#CCE5FF"

* "- fixedCosts: double"
* "- laborCosts: double"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ calculateNetRevenue()"
4. "+ decideExpenseApproval()"
5. "+ evaluate(): boolean"
6. "+ commit(): void"
7. "+ getState(): String[][]"

## 16. SimulationClock

stereotype: "«Utility»"

constraint: "{drives simulation}"

color: "#CCE5FF"

* "- currentTime: int"
* "- tickRate: int"

1. "+ advanceTime()"
2. "+ scheduleEvent()"
3. "+ getTime()"
4. "+ decideNextEventTrigger()"
5. "+ evaluate(): boolean"
6. "+ commit(): void"
7. "+ getState(): String[][]"

## 19. SimulationManager

stereotype: "«Simulation Kernel with Console & Waveform»"

constraint: "{drives simulation clk}"

color: "#CCE5FF"

1. "+ registerLoggables(objectName: String, logMap: Map<String, Supplier<Object>>)"
2. "+ logObject(objectName: String)"
3. "+ triggerLog(objectName: String)"
4. "+ loadInitFile(filename: String)"
5. "+ resetSimulation()"
6. "+ saveState(filename: String)"
7. "+ setVCD(filename: String)"
8. "+ run(int): void"
9. "+ stepOnce(): void"
10. "+ getClock(): SimulationClock"
11. "+ startConsole(): void"
12. "+ listObjects(): void"
13. "+ showObject(String): void"
14. "+ setValue(String, String, String): void"
15. "+ startLogging(String): void"
16. "+ stopLogging(): void"
17. "+ logTick(int): void"
18. "+ displayStates(): void"

## 1. Restaurant

stereotype: "«inherits SimEntity»"

constraint: "{manages operations}"

color: "#CCE5FF"

* "- name: String"
* "- tables: List<Table>"
* "- menu: List<Platter>"
* "- staff: List<Staff>"
* "- revenue: double"
* "- avgWalkTime: double"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ simulateDay()"
4. "+ decideStaffing()"
5. "+ decideTableAssignments()"
6. "+ decideInventoryRestock()"
7. "decideStaffing (uses Probability: RushHourOverloadChance)"
8. "decideInventoryRestock (uses Probability: IngredientSpoilageRate)"
9. "+ getGlobals(): GlobalVariables"
10. "+ getDecisionEngine(): GlobalMethods"
11. "+ evaluate(): boolean"
12. "+ commit(): void"
13. "+ getState(): String[][]"

## 2. Hostess

stereotype: "«inherits SimEntity»"

constraint: "{assigns tables}"

color: "#CCE5FF"

* "- name: String"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ greetCustomer(Customer)"
4. "+ seatCustomer(Customer, Table)"
5. "+ decideTableAssignment(Customer)"
6. "+ releaseTableDecision(Table, Customer)"
7. "+ decideSeating()"
8. "+ decideQueueOrder()"
9. "decideSeating (uses Probability: OptimalTableSelectionProbability)"
10. "decideQueueOrder (uses Probability: QueueBalkingProbability)"
11. "+ evaluate(): boolean"
12. "+ commit(): void"
13. "+ getState(): String[][]"

## 14. Stove

stereotype: "«inherits SimEntity»"

constraint: "{hot line}"

color: "#CCE5FF"

* "- slots: int"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ startCooking(Platter)"
4. "+ finishCooking(Platter)"
5. "+ advanceTime()"
6. "+ evaluate(): boolean"
7. "+ commit(): void"
8. "+ getState(): String[][]"

## 17. Oven

stereotype: "«inherits SimEntity»"

constraint: "{test appliance}"

color: "#CCE5FF"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ startCooking(Platter)"
4. "+ advanceTime()"
5. "+ evaluate(): boolean"
6. "+ commit(): void"
7. "+ getState(): String[][]"

## 18. Microwave

stereotype: "«inherits SimEntity»"

constraint: "{test appliance}"

color: "#CCE5FF"

1. "+ registerLoggables()"
2. "+ triggerLog()"
3. "+ startCooking(Platter)"
4. "+ advanceTime()"
5. "+ evaluate(): boolean"
6. "+ commit(): void"
7. "+ getState(): String[][]"