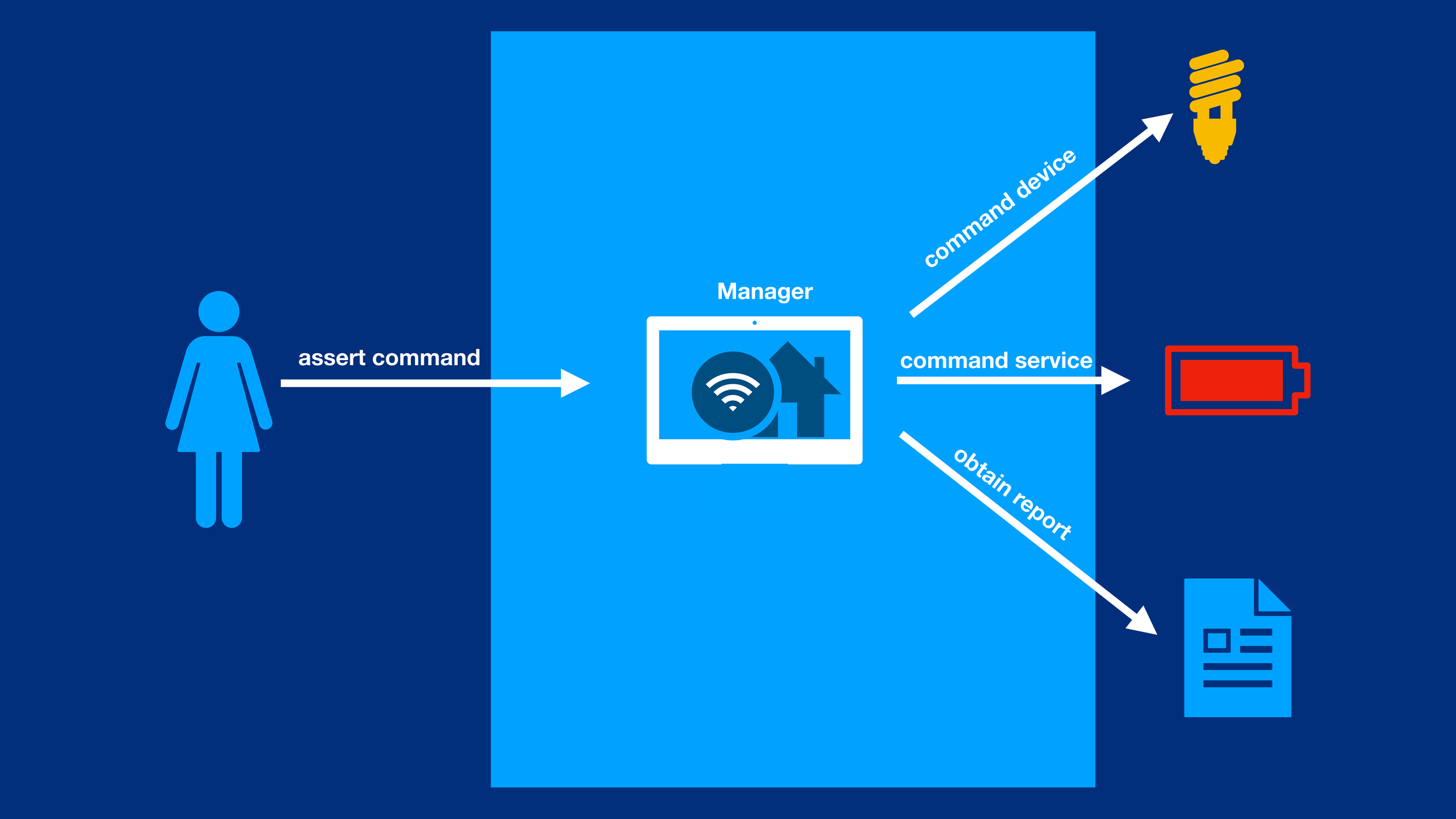
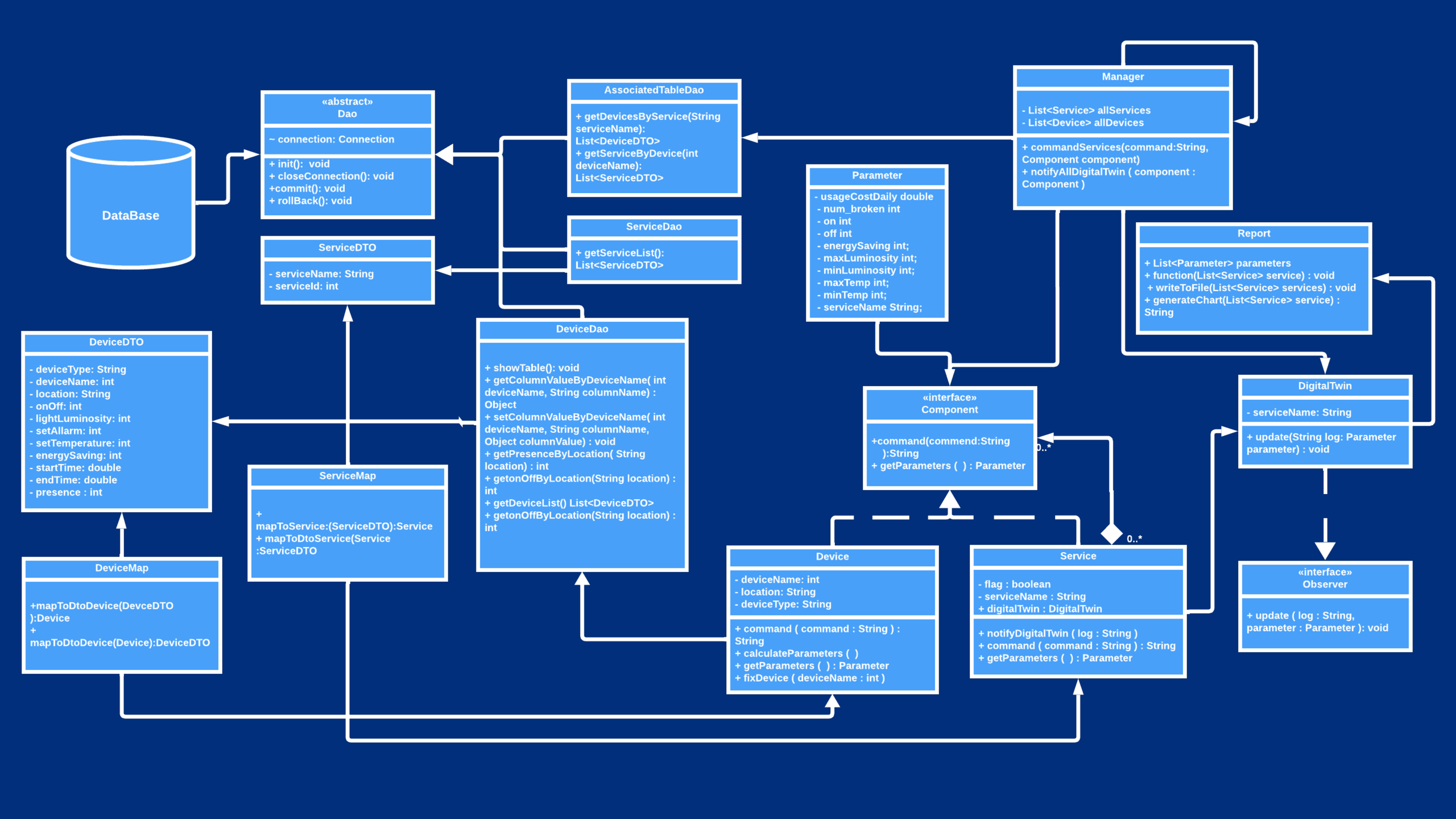
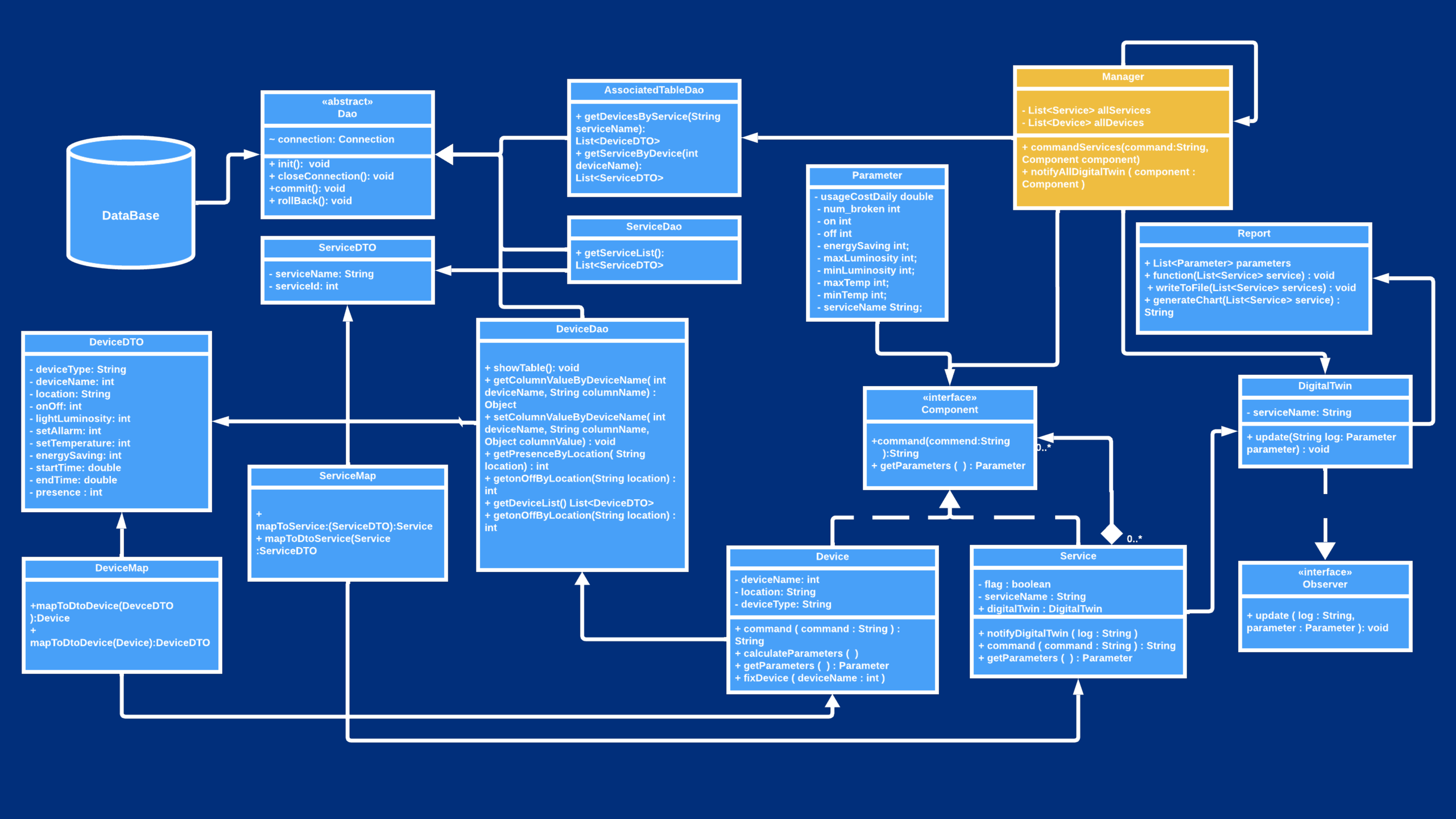
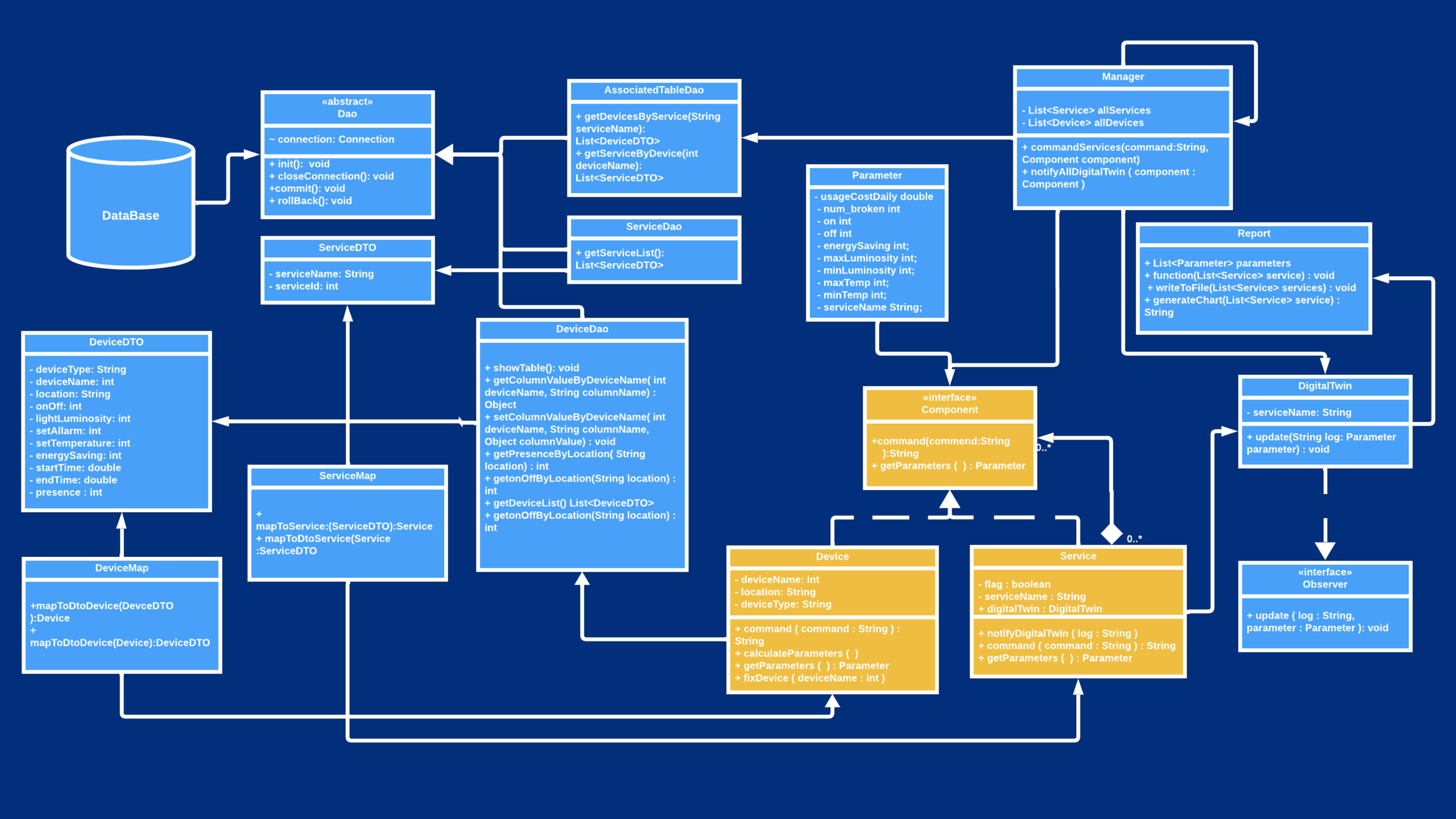
SmartHouse

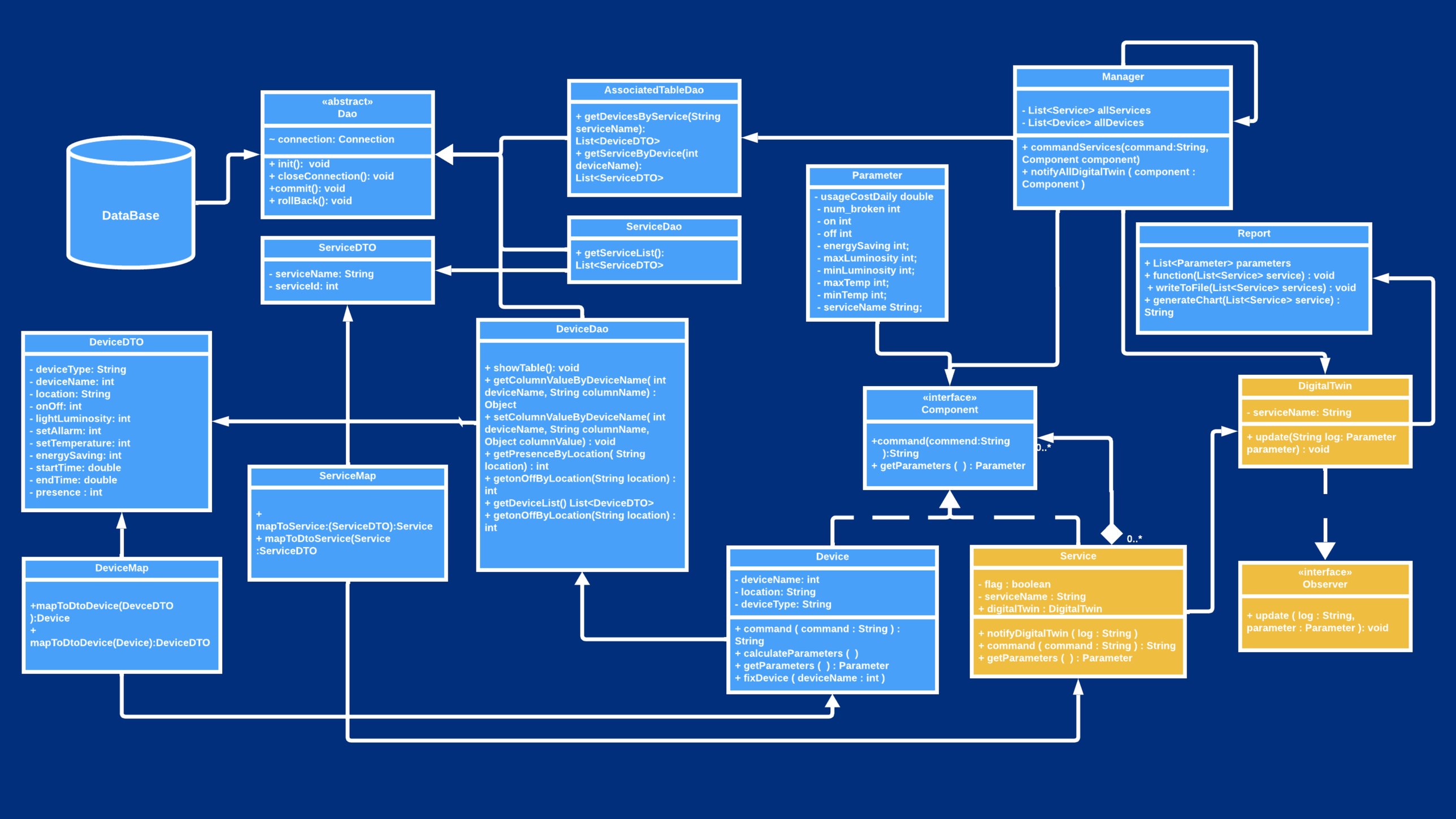
SmartHouse

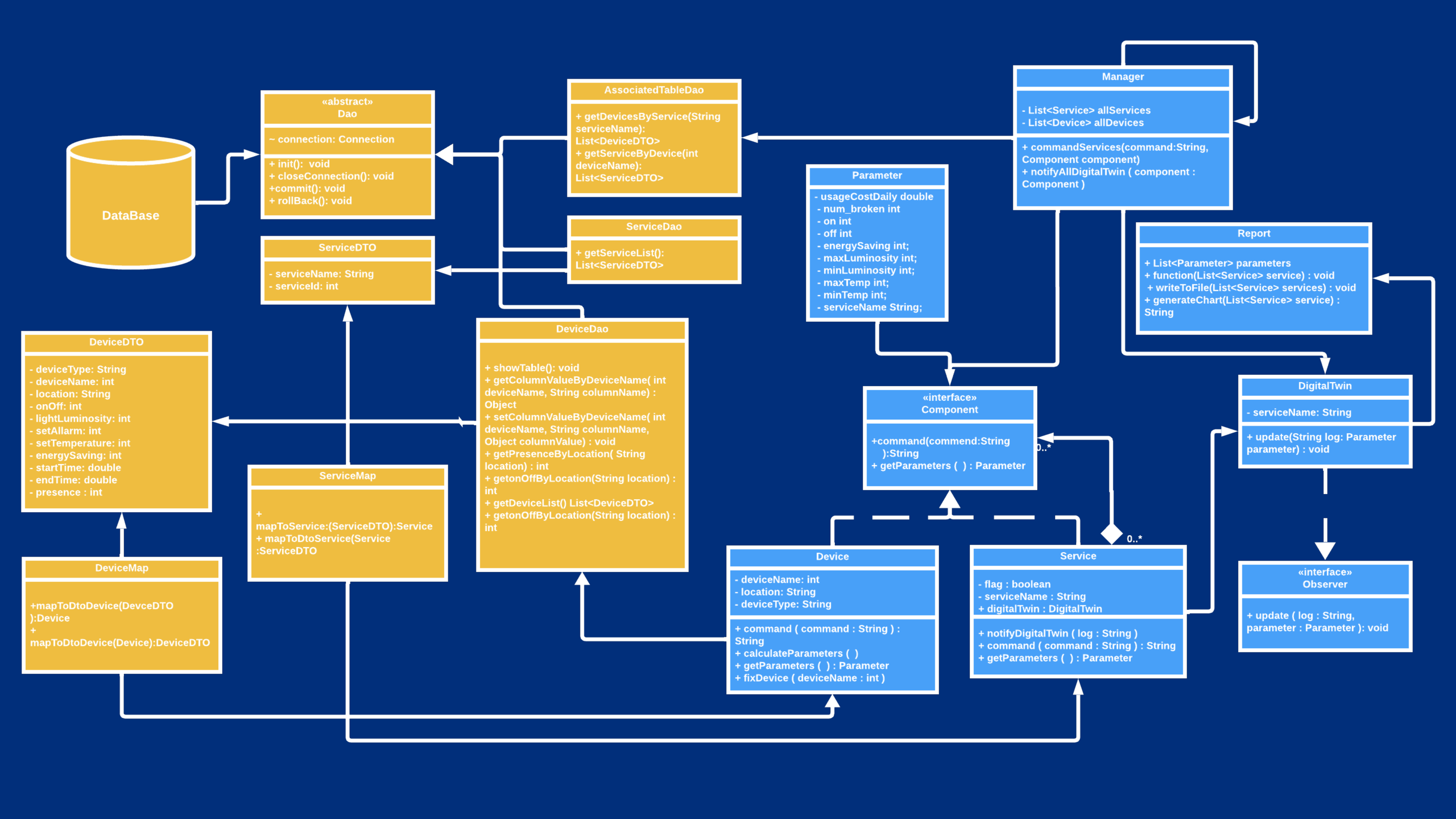

Manager
assert command
command device
command service
obtain report


DataBase
«abstract»
Dao
~ connection: Connection
+ init(): void
+ closeConnection(): void
+commit(): void
+ rollBack(): void
AssociatedTableDao
+ getDevicesByService(String serviceName): List<DeviceDTO>
+ getServiceByDevice(int deviceName): List<ServiceDTO>
ServiceDao
+ getServiceList(): List<ServiceDTO>
ServiceDTO
- serviceName: String
- serviceId: int
Manager
- List<Service> allServices
- List<Device> allDevices
+ commandServices(command:String, Component component)
+ notifyAllDigitalTwin (component : Component)
Parameter
- usageCostDaily double
- num_broken int
- on int
- off int
- energySaving int;
- maxLuminosity int;
- minLuminosity int;
- maxTemp int;
- minTemp int;
- serviceName String;
Report
+ List<Parameter> parameters
+ function(List<Service> service) : void
+ writeToFile(List<Service> services) : void
+ generateChart(List<Service> service) : String
DeviceDTO
- deviceType: String
- deviceName: int
- location: String
- onOff: int
- lightLuminosity: int
- setAllarm: int
- setTemperature: int
- energySaving: int
- startTime: double
- endTime: double
- presence : int
DeviceDao
+ showTable(): void
+ getColumnValueByDeviceName(int deviceName, String columnName) : Object
+ setColumnValueByDeviceName(int deviceName, String columnName, Object columnValue) : void
+ getPresenceByLocation(String location) : int
+ getonOffByLocation(String location) : int
+ getDeviceList() List<DeviceDTO>
+ getonOffByLocation(String location) : int
«interface»
Component
+command(commend:String):String
+ getParameters ( ) : Parameter
0..*
DigitalTwin
- serviceName: String
+ update(String log: Parameter parameter) : void
ServiceMap
+ mapToService:(ServiceDTO):Service
+ mapToDtoService(Service :ServiceDTO
DeviceMap
+mapToDtoDevice(DevceDTO):Device
+ mapToDtoDevice(Device):DeviceDTO
Device
- deviceName: int
- location: String
- deviceType: String
+ command (command : String) : String
+ calculateParameters ( )
+ getParameters ( ) : Parameter
+ fixDevice (deviceName : int)
Service
- flag : boolean
- serviceName : String
+ digitalTwin : DigitalTwin
+ notifyDigitalTwin (log : String)
+ command (command : String) : String
+ getParameters ( ) : Parameter
0..*
«interface»
Observer
+ update (log : String, parameter : Parameter): void


DataBase
«abstract»
Dao
~ connection: Connection
+ init(): void
+ closeConnection(): void
+commit(): void
+ rollBack(): void
AssociatedTableDao
+ getDevicesByService(String serviceName): List<DeviceDTO>
+ getServiceByDevice(int deviceName): List<ServiceDTO>
ServiceDao
+ getServiceList(): List<ServiceDTO>
ServiceDTO
- serviceName: String
- serviceId: int
Manager
- List<Service> allServices
- List<Device> allDevices
+ commandServices(command:String, Component component)
+ notifyAllDigitalTwin (component : Component)
Parameter
- usageCostDaily double
- num_broken int
- on int
- off int
- energySaving int;
- maxLuminosity int;
- minLuminosity int;
- maxTemp int;
- minTemp int;
- serviceName String;
Report
+ List<Parameter> parameters
+ function(List<Service> service) : void
+ writeToFile(List<Service> services) : void
+ generateChart(List<Service> service) : String
DeviceDTO
- deviceType: String
- deviceName: int
- location: String
- onOff: int
- lightLuminosity: int
- setAllarm: int
- setTemperature: int
- energySaving: int
- startTime: double
- endTime: double
- presence : int
DeviceDao
+ showTable(): void
+ getColumnValueByDeviceName(int deviceName, String columnName) : Object
+ setColumnValueByDeviceName(int deviceName, String columnName, Object columnValue) : void
+ getPresenceByLocation(String location) : int
+ getonOffByLocation(String location) : int
+ getDeviceList() List<DeviceDTO>
+ getonOffByLocation(String location) : int
«interface»
Component
+command(commend:String):String
+ getParameters ( ) : Parameter
0..*
DigitalTwin
- serviceName: String
+ update(String log: Parameter parameter) : void
ServiceMap
+ mapToService:(ServiceDTO):Service
+ mapToDtoService(Service :ServiceDTO
DeviceMap
+mapToDtoDevice(DevceDTO):Device
+ mapToDtoDevice(Device):DeviceDTO
Device
- deviceName: int
- location: String
- deviceType: String
+ command (command : String) : String
+ calculateParameters ( )
+ getParameters ( ) : Parameter
+ fixDevice (deviceName : int)
Service
- flag : boolean
- serviceName : String
+ digitalTwin : DigitalTwin
+ notifyDigitalTwin (log : String)
+ command (command : String) : String
+ getParameters ( ) : Parameter
0..*
«interface»
Observer
+ update (log : String, parameter : Parameter): void


DataBase
«abstract»
Dao
~ connection: Connection
+ init(): void
+ closeConnection(): void
+commit(): void
+ rollBack(): void
AssociatedTableDao
+ getDevicesByService(String serviceName): List<DeviceDTO>
+ getServiceByDevice(int deviceName): List<ServiceDTO>
ServiceDao
+ getServiceList(): List<ServiceDTO>
Manager
- List<Service> allServices
- List<Device> allDevices
+ commandServices(command:String, Component component)
+ notifyAllDigitalTwin (component : Component)
ServiceDTO
- serviceName: String
- serviceId: int
Parameter
- usageCostDaily double
- num_broken int
- on int
- off int
- energySaving int;
- maxLuminosity int;
- minLuminosity int;
- maxTemp int;
- minTemp int;
- serviceName String;
Report
+ List<Parameter> parameters
+ function(List<Service> service) : void
+ writeToFile(List<Service> services) : void
+ generateChart(List<Service> service) : String
DeviceDTO
- deviceType: String
- deviceName: int
- location: String
- onOff: int
- lightLuminosity: int
- setAllarm: int
- setTemperature: int
- energySaving: int
- startTime: double
- endTime: double
- presence : int
DeviceDao
+ showTable(): void
+ getColumnValueByDeviceName(int deviceName, String columnName) : Object
+ setColumnValueByDeviceName(int deviceName, String columnName, Object columnValue) : void
+ getPresenceByLocation(String location) : int
+ getonOffByLocation(String location) : int
+ getDeviceList() List<DeviceDTO>
+ getonOffByLocation(String location) : int
«interface»
Component
+command(commend:String):String
+ getParameters ( ) : Parameter
0..*
DigitalTwin
- serviceName: String
+ update(String log: Parameter parameter) : void
ServiceMap
+ mapToService:(ServiceDTO):Service
+ mapToDtoService(Service :ServiceDTO
DeviceMap
+mapToDtoDevice(DevceDTO):Device
+ mapToDtoDevice(Device):DeviceDTO
Device
- deviceName: int
- location: String
- deviceType: String
+ command (command : String) : String
+ calculateParameters ( )
+ getParameters ( ) : Parameter
+ fixDevice (deviceName : int)
Service
- flag : boolean
- serviceName : String
+ digitalTwin : DigitalTwin
+ notifyDigitalTwin (log : String)
+ command (command : String) : String
+ getParameters ( ) : Parameter
0..*
«interface»
Observer
+ update (log : String, parameter : Parameter): void


DataBase
«abstract»
Dao
~ connection: Connection
+ init(): void
+ closeConnection(): void
+commit(): void
+ rollBack(): void
AssociatedTableDao
+ getDevicesByService(String serviceName): List<DeviceDTO>
+ getServiceByDevice(int deviceName): List<ServiceDTO>
ServiceDao
+ getServiceList(): List<ServiceDTO>
ServiceDTO
- serviceName: String
- serviceId: int
Manager
- List<Service> allServices
- List<Device> allDevices
+ commandServices(command:String, Component component)
+ notifyAllDigitalTwin (component : Component)
Parameter
- usageCostDaily double
- num_broken int
- on int
- off int
- energySaving int;
- maxLuminosity int;
- minLuminosity int;
- maxTemp int;
- minTemp int;
- serviceName String;
Report
+ List<Parameter> parameters
+ function(List<Service> service) : void
+ writeToFile(List<Service> services) : void
+ generateChart(List<Service> service) : String
DeviceDTO
- deviceType: String
- deviceName: int
- location: String
- onOff: int
- lightLuminosity: int
- setAllarm: int
- setTemperature: int
- energySaving: int
- startTime: double
- endTime: double
- presence : int
DeviceDao
+ showTable(): void
+ getColumnValueByDeviceName(int deviceName, String columnName) : Object
+ setColumnValueByDeviceName(int deviceName, String columnName, Object columnValue) : void
+ getPresenceByLocation(String location) : int
+ getonOffByLocation(String location) : int
+ getDeviceList() List<DeviceDTO>
+ getonOffByLocation(String location) : int
«interface»
Component
+command(commend:String):String
+ getParameters ( ) : Parameter
0..*
DigitalTwin
- serviceName: String
+ update(String log: Parameter parameter) : void
ServiceMap
+ mapToService:(ServiceDTO):Service
+ mapToDtoService(Service :ServiceDTO
DeviceMap
+mapToDtoDevice(DevceDTO):Device
+ mapToDtoDevice(Device):DeviceDTO
Device
- deviceName: int
- location: String
- deviceType: String
+ command (command : String) : String
+ calculateParameters ( )
+ getParameters ( ) : Parameter
+ fixDevice (deviceName : int)
Service
- flag : boolean
- serviceName : String
+ digitalTwin : DigitalTwin
+ notifyDigitalTwin (log : String)
+ command (command : String) : String
+ getParameters ( ) : Parameter
0..*
«interface»
Observer
+ update (log : String, parameter : Parameter): void


DataBase
«abstract»
Dao
~ connection: Connection
+ init(): void
+ closeConnection(): void
+commit(): void
+ rollBack(): void
AssociatedTableDao
+ getDevicesByService(String serviceName): List<DeviceDTO>
+ getServiceByDevice(int deviceName): List<ServiceDTO>
ServiceDao
+ getServiceList(): List<ServiceDTO>
ServiceDTO
- serviceName: String
- serviceId: int
Manager
- List<Service> allServices
- List<Device> allDevices
+ commandServices(command:String, Component component)
+ notifyAllDigitalTwin (component : Component)
Parameter
- usageCostDaily double
- num_broken int
- on int
- off int
- energySaving int;
- maxLuminosity int;
- minLuminosity int;
- maxTemp int;
- minTemp int;
- serviceName String;
Report
+ List<Parameter> parameters
+ function(List<Service> service) : void
+ writeToFile(List<Service> services) : void
+ generateChart(List<Service> service) : String
DeviceDTO
- deviceType: String
- deviceName: int
- location: String
- onOff: int
- lightLuminosity: int
- setAllarm: int
- setTemperature: int
- energySaving: int
- startTime: double
- endTime: double
- presence : int
DeviceDao
+ showTable(): void
+ getColumnValueByDeviceName(int deviceName, String columnName) : Object
+ setColumnValueByDeviceName(int deviceName, String columnName, Object columnValue) : void
+ getPresenceByLocation(String location) : int
+ getonOffByLocation(String location) : int
+ getDeviceList() List<DeviceDTO>
+ getonOffByLocation(String location) : int
«interface»
Component
+command(commend:String):String
+ getParameters ( ) : Parameter
0..*
DigitalTwin
- serviceName: String
+ update(String log: Parameter parameter) : void
ServiceMap
+ mapToService:(ServiceDTO):Service
+ mapToDtoService(Service :ServiceDTO
DeviceMap
+mapToDtoDevice(DevceDTO):Device
+ mapToDtoDevice(Device):DeviceDTO
Device
- deviceName: int
- location: String
- deviceType: String
+ command (command : String) : String
+ calculateParameters ( )
+ getParameters ( ) : Parameter
+ fixDevice (deviceName : int)
Service
- flag : boolean
- serviceName : String
+ digitalTwin : DigitalTwin
+ notifyDigitalTwin (log : String)
+ command (command : String) : String
+ getParameters ( ) : Parameter
0..*
«interface»
Observer
+ update (log : String, parameter : Parameter): void

Implemented Technologies:

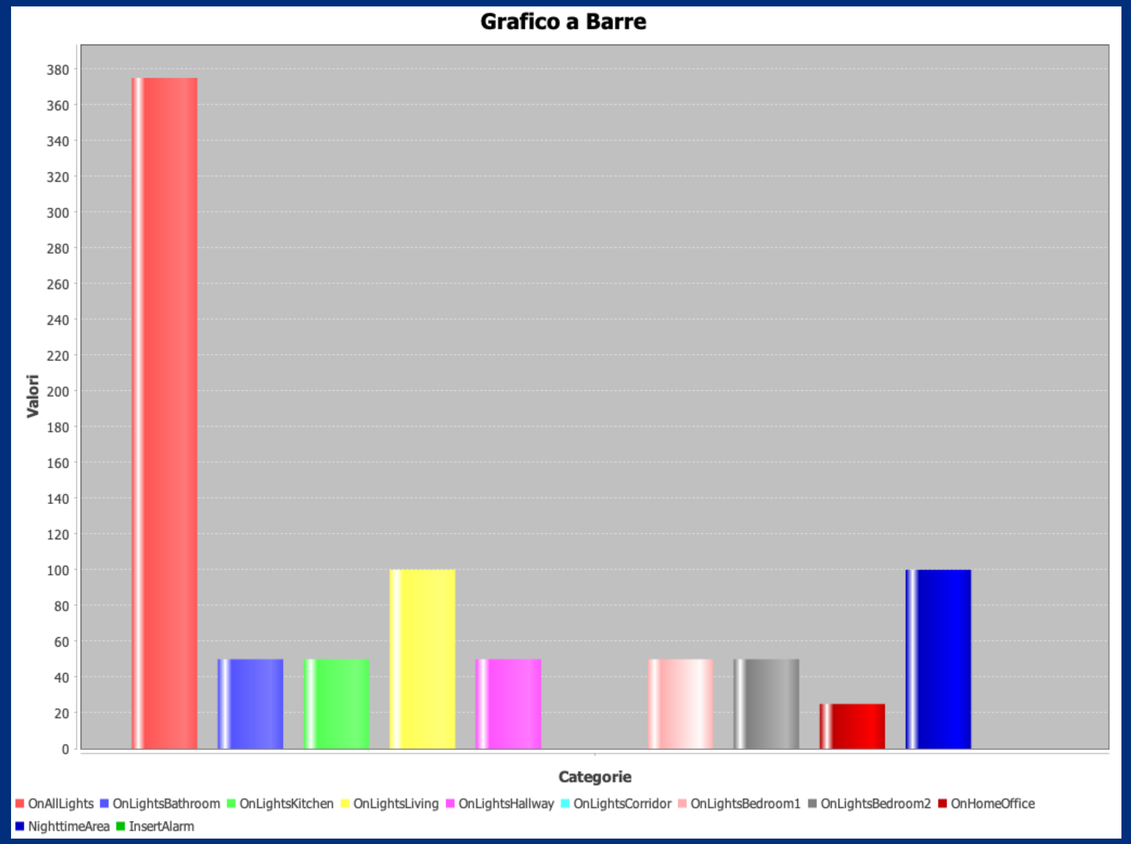
Maven: easier libraries importation

sqLite: storage devices ‘ information

JUnit : testing framework

Honorable mentions: log4J, lambda functions, lombok and jfreechart

Results:




```
Service: OnHomeOffice,  Device: 16 numTimeBrokenDevice: 0
Service: OnHomeOffice,  Device: 17 numTimeBrokenDevice: 0
Service: NighttimeArea,  Device: 11 numTimeBrokenDevice: 1
Service: NighttimeArea,  Device: 12 numTimeBrokenDevice: 0
Service: NighttimeArea,  Device: 13 numTimeBrokenDevice: 0
Service: NighttimeArea,  Device: 14 numTimeBrokenDevice: 0
Service: NighttimeArea,  Device: 15 numTimeBrokenDevice: 0
Service: InsertAlarm,  Device: 23 numTimeBrokenDevice: 0
Service: InsertAlarm,  Device: 24 numTimeBrokenDevice: 0
Service: InsertAlarm,  Device: 25 numTimeBrokenDevice: 0
Service: InsertAlarm,  Device: 26 numTimeBrokenDevice: 0
Service: InsertAlarm,  Device: 27 numTimeBrokenDevice: 0
Service: OnAllLights,  Device: 1 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 2 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 3 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 4 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 5 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 6 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 7 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 8 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 9 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 10 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 11 numTimegetDeviceOn: 0
Service: OnAllLights,  Device: 12 numTimegetDeviceOn: 2
Service: OnAllLights,  Device: 13 numTimegetDeviceOn: 2
Service: OnAllLights,  Device: 14 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 15 numTimegetDeviceOn: 1
Service: OnAllLights,  Device: 16 numTimegetDeviceOn: 0
Service: OnAllLights,  Device: 17 numTimegetDeviceOn: 1
Service: OnLightsBathroom,  Device: 1 numTimegetDeviceOn: 1
Service: OnLightsBathroom,  Device: 3 numTimegetDeviceOn: 1
Service: OnLightsKitchen,  Device: 2 numTimegetDeviceOn: 1
Service: OnLightsKitchen,  Device: 4 numTimegetDeviceOn: 1
Service: OnLightsLiving,  Device: 5 numTimegetDeviceOn: 1
```