

<ServiceWolf> Design Proposal

Document Author(s): Michael Golden

Date: 02/20/21

Design Rationale

The two most essential objects in this design are ServiceGroup and Incident. A ServiceGroup is an organization responsible for overseeing the resolution of assigned instances. An Incident is a request made by a user that needs assistance of a ServiceGroup. Both objects are important to the system because the purpose of the program is to manage the status(state) of an Incident as the ServiceGroup works to resolve it.

Each ServiceGroup has a name and a SortedList of Incidents. The list is sorted numerically by Incident ID. Each Incident has a title, caller (person making the request for help), description, and a Unique ID number.

An Incident or ServiceGroup can be added, deleted, and edited through the ServiceWolf class. Editing an Incident is dependent on the current State of the incident.

The ServiceWolfGUI class handles the user interface and has 1 instance of ServiceWolf. ServiceWolf can have many instances of ServiceGroup. This is because there can be multiple ServiceGroups working in the same system or for the same company. Each ServiceGroup has many instances of Incident. Multiple Incidents are stored in a SortedList of the ServiceGroup that it is assigned to. Each Incident is in one of 5 states; New, InProgress, OnHold, Resolved, and Cancelled. Each Incident can access these states through its relationship with the State Interface. Each of the 5 states implements the State Interface. The states control the functionality of moving the Incident through the ServiceWolf FSM.

The State Pattern is used in this design in order to model the states as objects in their own classes instead of as conditional statements. Using this design pattern is appropriate because the state of the Incident controls the Incident's behavior.

A limitation of this design is that ServiceWolf cannot directly access an Incident. It can only access an Incident through ServiceGroup. This design may hinder the ability to add new roles in the future because of the linear composition relationship from ServiceWolf to ServiceGroup to Incident. However this design would make it easy to add new states in the future.

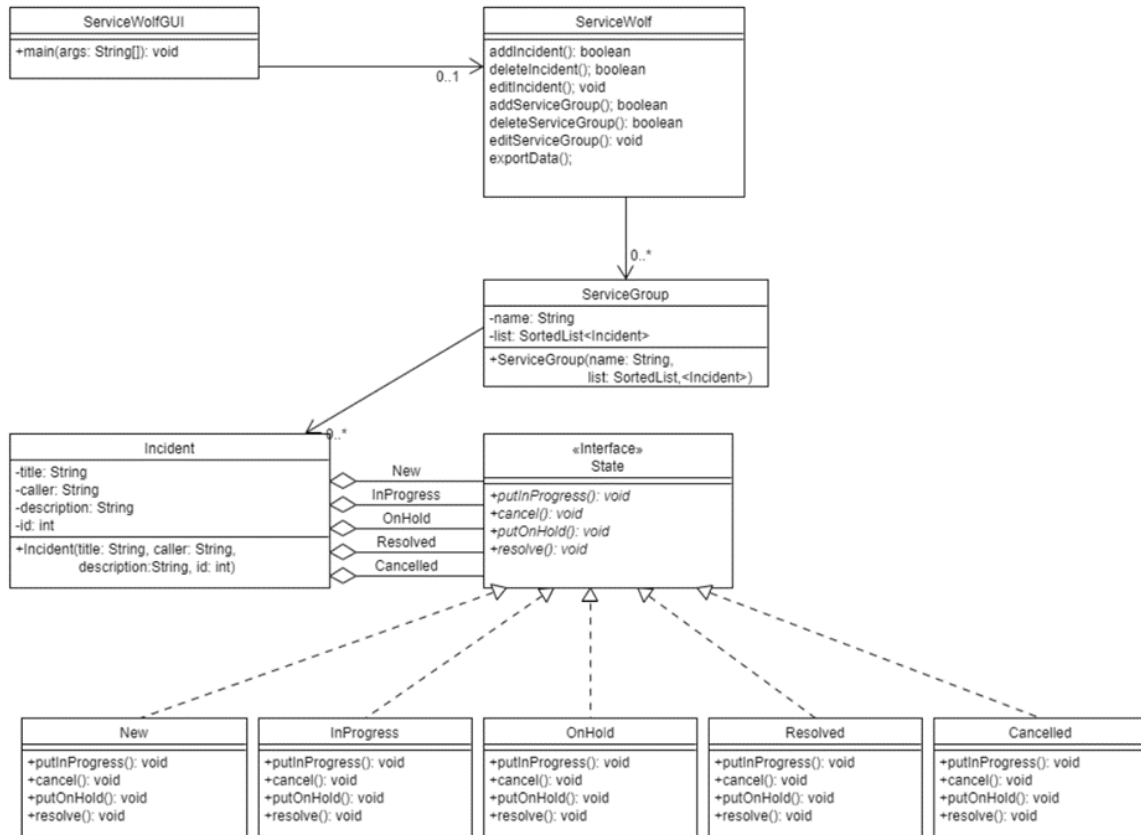


Figure 1: Class Diagram for <ServiceWolf>

Document Revision History

Date	Author	Change Description
02/20/21	Michael Golden	• Created Document
02/22/21	Michael Golden	• Added Figure 1
02/24/21	Michael Golden	• Added Design Rationale