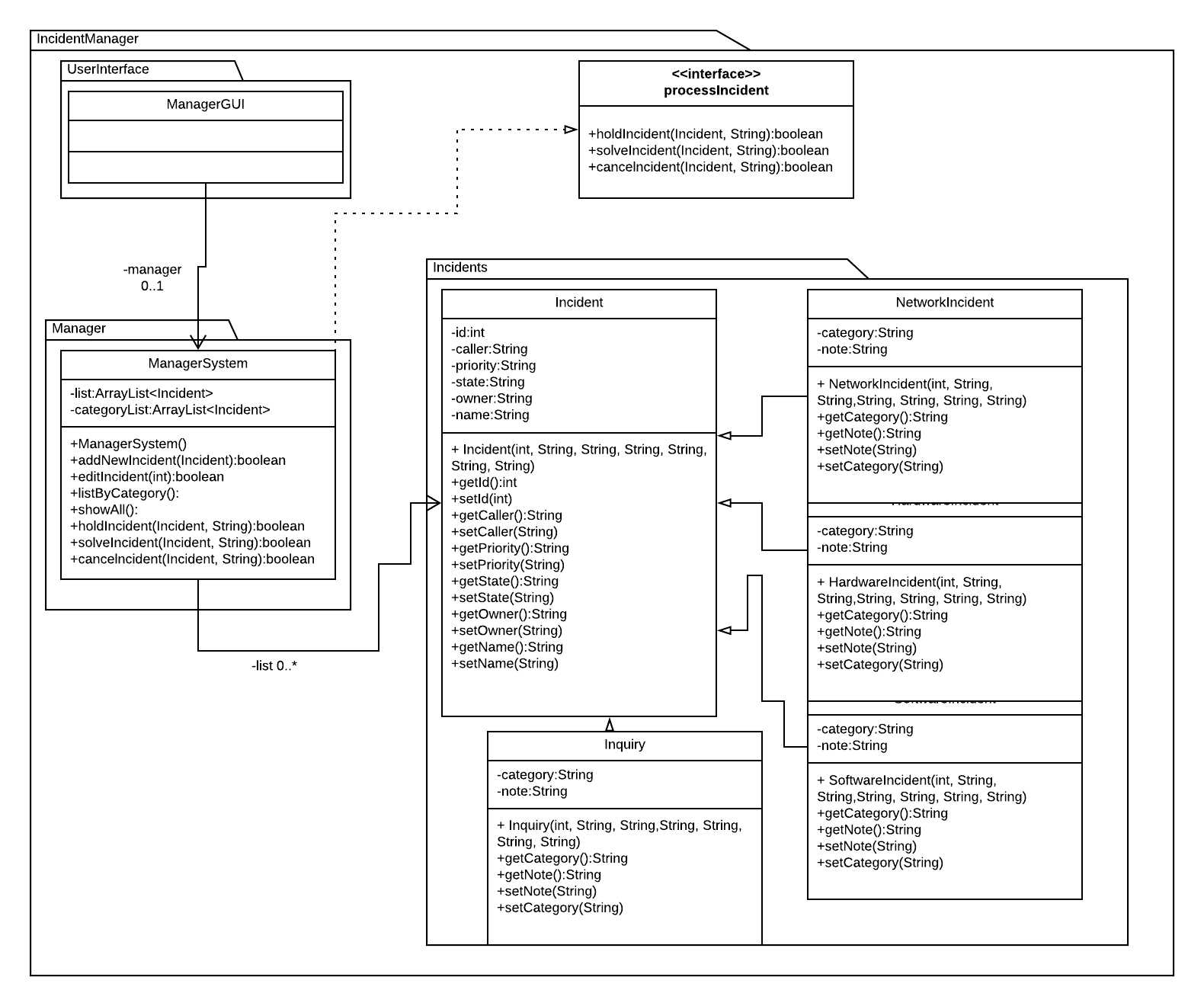
**IncidentManager Design Proposal**

**Document Author(s): Max Richgruber**

**Date: 10/26/18**

**Design Rationale**

For the *IncidentManager* program, I decided to have the core of the program run through the *Manager* class, which keeps track of all the currently stored incidents and can interact with them. The *Manager* can do this because it has multiple instances of *Incident* objects stored in an array list, functioning as a database of sorts for all of the incidents in the system. The *Incident* objects exist in four types*: NetworkIncident, HardwareIncident, SoftwareIncident*, and *Inquiry* which all extend the Incident superclass that is used by the Manager. The Incidents are individually added and created by the user to *Manager.* Once an Incident has been created it can be processed with the *processIncident* interface, which allows for three different types of processing: holding, solving, or canceling. The *IncidentManagerUI* facilitates the interaction between the program’s user and the *Manager* class.



**Figure 1: Class Diagram for IncidentManager**

**Document Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Change Description** |
| **Oct 2018** | Max Richgruber | * Added UML and rationale |