

## Project 2: Ticket Tracker - Design Proposal

---

**Document Author(s):** Nicholas Board

**Date:** 10/21/2016

---

### Design Rationale

I believe that our team has created an optimal design for EcoResearch's IT crew when dealing with request for help with the Ecosystem Simulator and other programs. The design utilizes the concept of a Finite State Machine, using the state pattern design. A ticket can be in multiple different states, each represented by a different class and a method for each transition. The state pattern is appropriate because, no matter what, each ticket MUST be in one of these states, and can only follow the exact transitions that are define in the program. This is facilitated by a StateController class that interacts with a ticket item. The tickets are kept in a TicketList object that allows the user to filter and show all the tickets. Each Ticket object has a NoteList object, and is in turn made up of NoteItems. The TicketList and NoteList have similar behavior, so they both abstract from an ItemList class that allows the basic behaviors of adding, deleting, and editing. It is important to have a TicketList class because you need a way to organize all the tickets into one set. Similarly, for the NoteList class, the difference being that the NoteList is constructed inside of individual Ticket objects. And of course, you need stand-alone Ticket and Note classes so you can construct each individual ticket and note objects that the lists will store. These objects represent the model of the project.

The actual TicketList is contained in an XMLReader class. This class controls the input and output into the file system. The user can load previous sets of tickets, as well as save the one they are working on. The user can also use the XMLReader to start a new set of tickets, and quit out of the application. No preexisting data is needed to use the program, unless you want to load already existing TicketList. XMLReader's functionality is essential to the program being useful in terms of accessing existing data and creating existing data. The XMLReader class is the controller of the project. The main method is contained in a GUI class that will be the view of the program, and is very user-friendly.

The UML class diagram for the design is in an attached image file.

---

### Document Revision History

Date	Author	Change Description
10/21/16	ndboard	<ul style="list-style-type: none"><li>Started design proposal</li><li>Finished design proposal</li></ul>