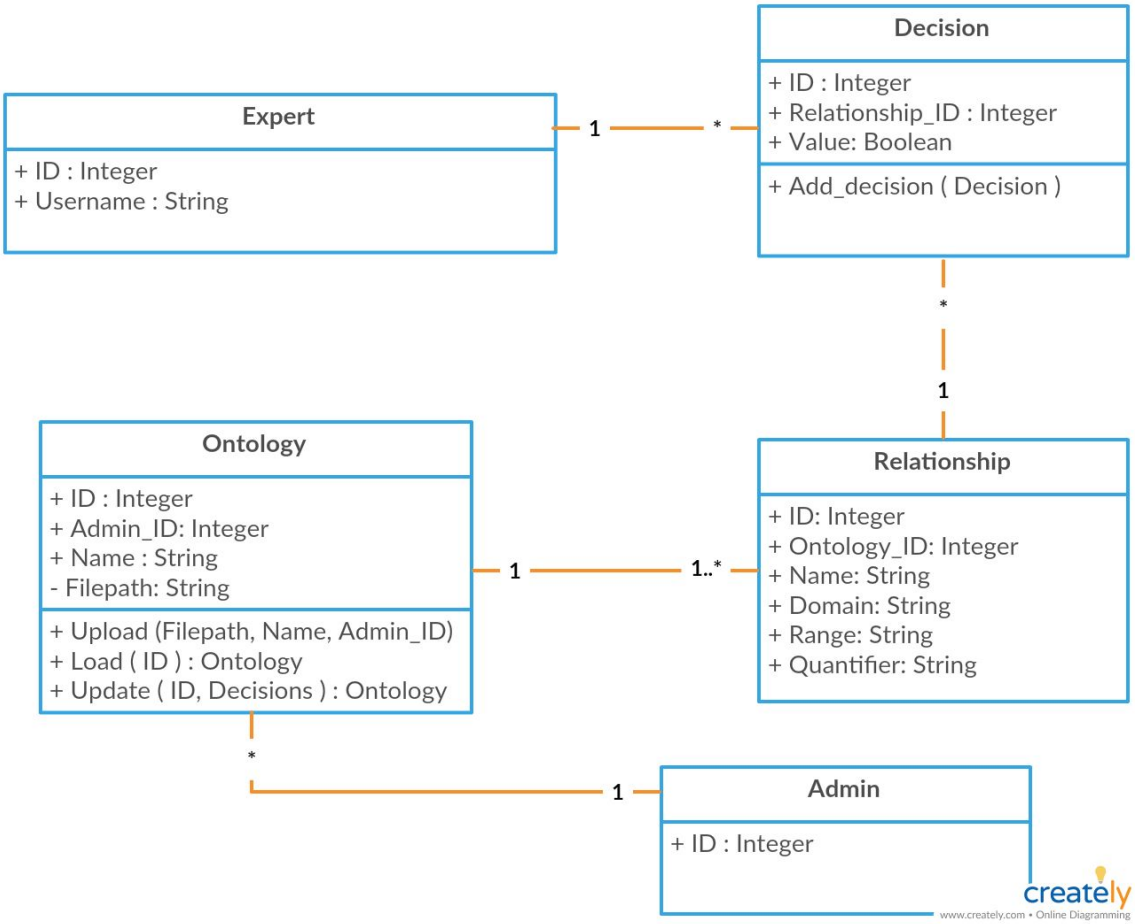
Product Design

Team 22, **Anirudha**, **Saujas**, **Shikhar**

Design Model

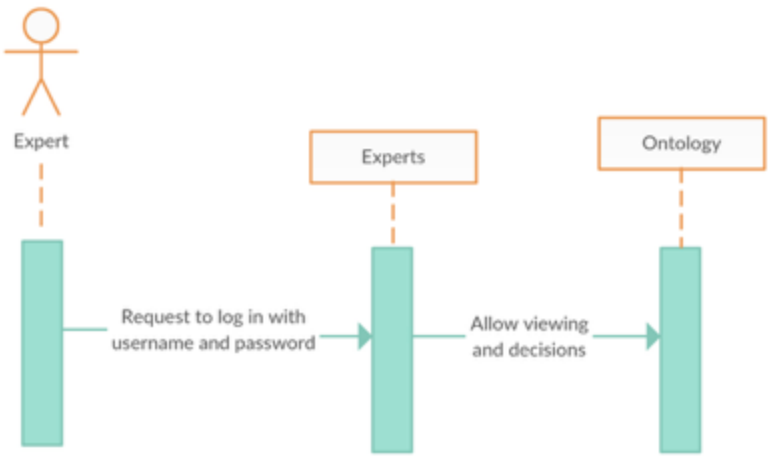


Expert	*Class state* • *ID: Unique identifier* • *Username: Identifier for authentication*

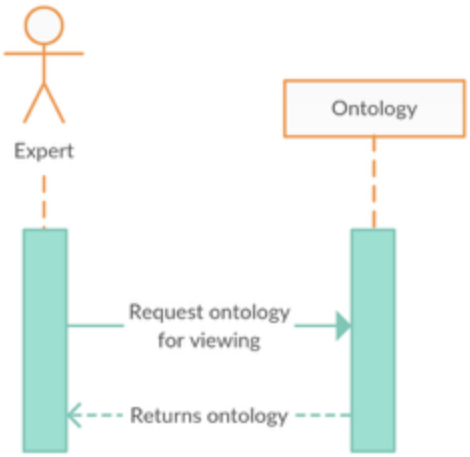
Ontology	*Class state* • *ID: Unique identifier* • *Admin_ID: ID of owning admin* • *Name: Name of ontology for viewing* *Class behavior* • *Upload: Add new ontology on server* • *Load: Retrieve ontology for viewing* • *Update: Generate new ontology file based on decisions*
Relationship	*Class state* • *ID: Unique identifier* • *Ontology_ID: ID of ontology to which the relationship belongs* • *Name: Name of relationship* • *Domain: Domain on which relationship is defined* • *Range: Range of values the relationship can relate* • *Quantifier: Type of restriction on quantities*
Decision	*Class state* • *ID: Unique identifier* • *Relationship_ID: ID of relationship which is decided* • *Value: Boolean indicating whether the decision was accepted* *Class behavior* • *Add_decision: Add a new decision to the database*
Admin	*Class state* • *ID: Unique identifier*

Sequence Diagram(s)

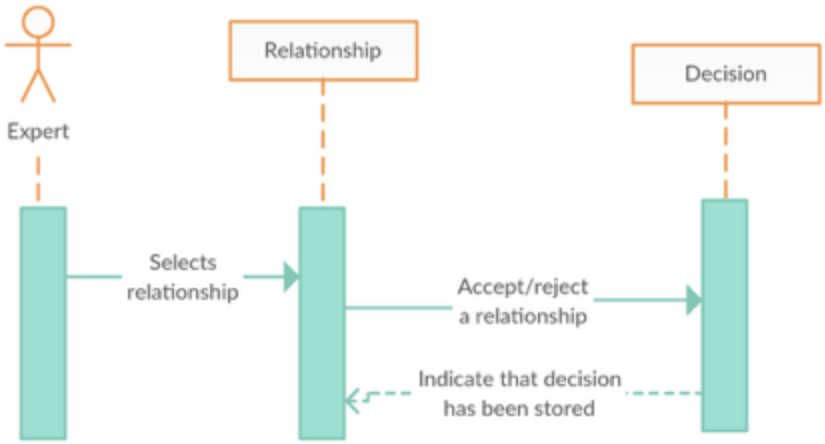
Authenticate:



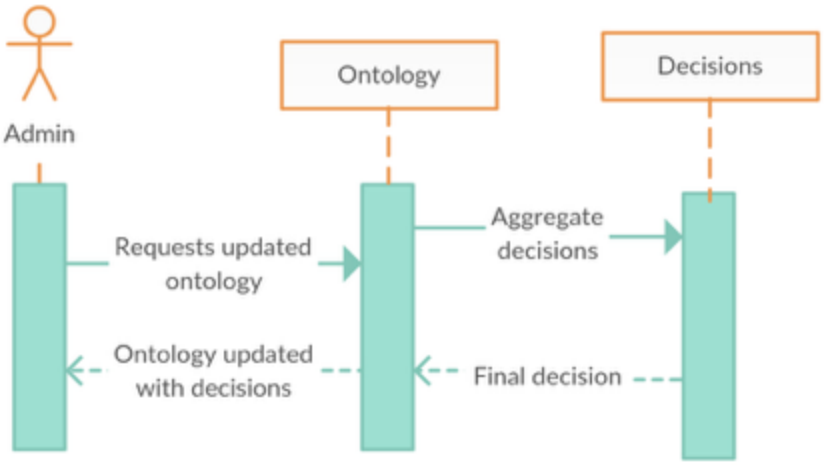
Load and Visualise:



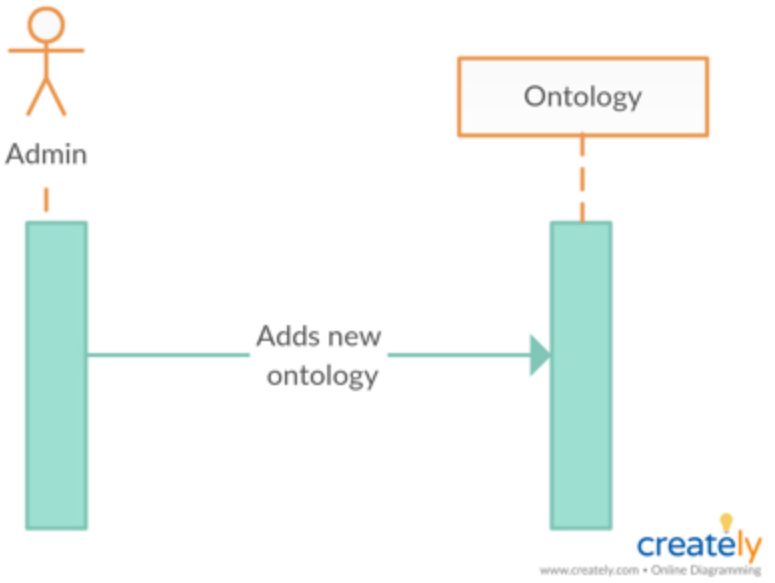
Accept/Reject:



Get verified ontology:



Store ontology:



Design Rationale

We had an option between two visualisation softwares - OntoGraf for Protege, and WebVOWL. OntoGraf is a plugin for a desktop software, and if it had to be used, it would have to be made compatible with a web application. WebVOWL was a standalone visualisation package, and an ontology management system had to be built. We decided to use WebVOWL since we believed a good visualisation system would be more difficult to build than an ontology management system.