

# Document Review

Design Document

October 22, 2019

## 1 Roles

Moderator: Evan

Reviewers: Braunson, Clinton

Reviewees: Camille, Evan

## 2 Objectives

A review of the design document was conducted remotely over a video call. Preliminary document was read by all reviewers who compiled their notes and shared them with the authors. The authors were able to answer all our questions and provide positive feedback to defects raised. The design document was improved following this review. The majority of the defects will be resolved for ID2 and the rest will be filed on Github issues for later resolution.

## 3 Scoring

Provide a rating from 1 - 5.

1 - Work has room for improvement

3 - Work was done to an acceptable standard

5 - The work went above and beyond expectations

Criteria	Score
Completeness: (completed to the full scope of the requirements)	4
Concise: (the purpose is achieved without undue complication)	5
Structure/organization: (does the work flow, are components grouped logically?)	4
Grammar:	4

## 4 Notes

What was done well?

- The document preamble was well written and accurate

- The document was expansive, covering all methods and fields discussed in brainstorming sessions
- The document was written in "one voice"
- The design was clearly well planned and thought out for future expansion

**Other notes:**

- Document is subject to change as the project develops
- The parser may be split up as more functionality is supported
- To enable further expansion of the logic core of our application (the parser) it maybe necessary to break the parser into an interface

## 5 List of Defects

- Table of contents would be nice to have
- The constructor types are missing for the Parser
- A visual diagram would help users understand the pipes and filter flow of data
- Lacking a description in removePredicate, the redraw being handled here may cause a cyclic dependency between the timeline and its filters.
- Predicates being stored as named strings could make it easier for users to remove or hide some of their filters
- Lack a method to get all predicates from a filter class
- Lack a method to toggle a filter
- Types should be well defined following typescript types where possible and more explicitly when required (ie, does a number type allow only integers or also floats?)
- UML should flow downwards starting from the parser
- UML has connections that are not described in the method types or fields in the class
- UML is missing fields that are described in the design document