It optimizes the connection to the datastore using caching, and other capabilities to be added over time. It should provide a local observer design pattern to let the rest of the application know when there are updates. Ideally, also eventually implement observer design pattern against the server backend, DatabaseManager manages the connection to the datastore so polling isn't needed. NOTE: The local in-memory cache will allow the application to refresh as much as it want's, and not worry about db impact, or user experience impact.

NOTE: Since DatabaseManager is doing caching, when a method is called that returns life, a reference to data that is cached, the cache (maybe in a dictionary) should be searched so that if that object has already been created and a reference provided before, that same reference is provided again.

connection

Get data from the datastore
Add data to the datastore
Modify data in the data store
When an object is to be returned from a function, that is backed by
the database, a reference should be returned. Connection defines the methods the application can use to:

NOTE: In order to have just one location to define what type of connection to use, Connection will be mostly abstract, except for its default constructor. The default constructor will implemented code to return the connection type the developer wants to be active. We may move the definition into a config file, but that isn't required.

CsvConnection:CsvConnection(QString pathToCsvFolder) JsonConnection.JsonConnection(QString pathToJson) ...implement abstract methods from parent() ...implement abstract methods from parent() CsvConnection JsonConnection Examples of options, probaby won't implement all of these. pathToCsvFolder -pathToJson NOTE: Since sqlite and mysql should have some decent overlap ideally things that can be shared are implemented here void:getTvShow(QString name, TvShow& tvshow)
User:getUser(QString username)
addReactionToEpisode(Episode episode, Reaction reaction)
markEpisodeAsWatched(Episode episode) ...implement abstract methods from parent() Connection SqlConnection MySqlConnection:MySqlConnection(QString host, quint port, QString username, QString password, QString database) .. etc() \'\ TvShow&:getTvShow(QString name)
The above pattern for getTvShow will be repeated to allign with iconnection interface() MySqlConnection DatabaseManager:DatabaseManager(Connection connection) DatabaseManager One Connection only Singleton SqlLiteConnection pathToSq3 -quint port -QString username -QString password **QString** database -QString host

NOTE: Unless there are security concerns, there will most likely be just a single user connection to the db, used by the application, this is not the same as the user accounts for each end user.