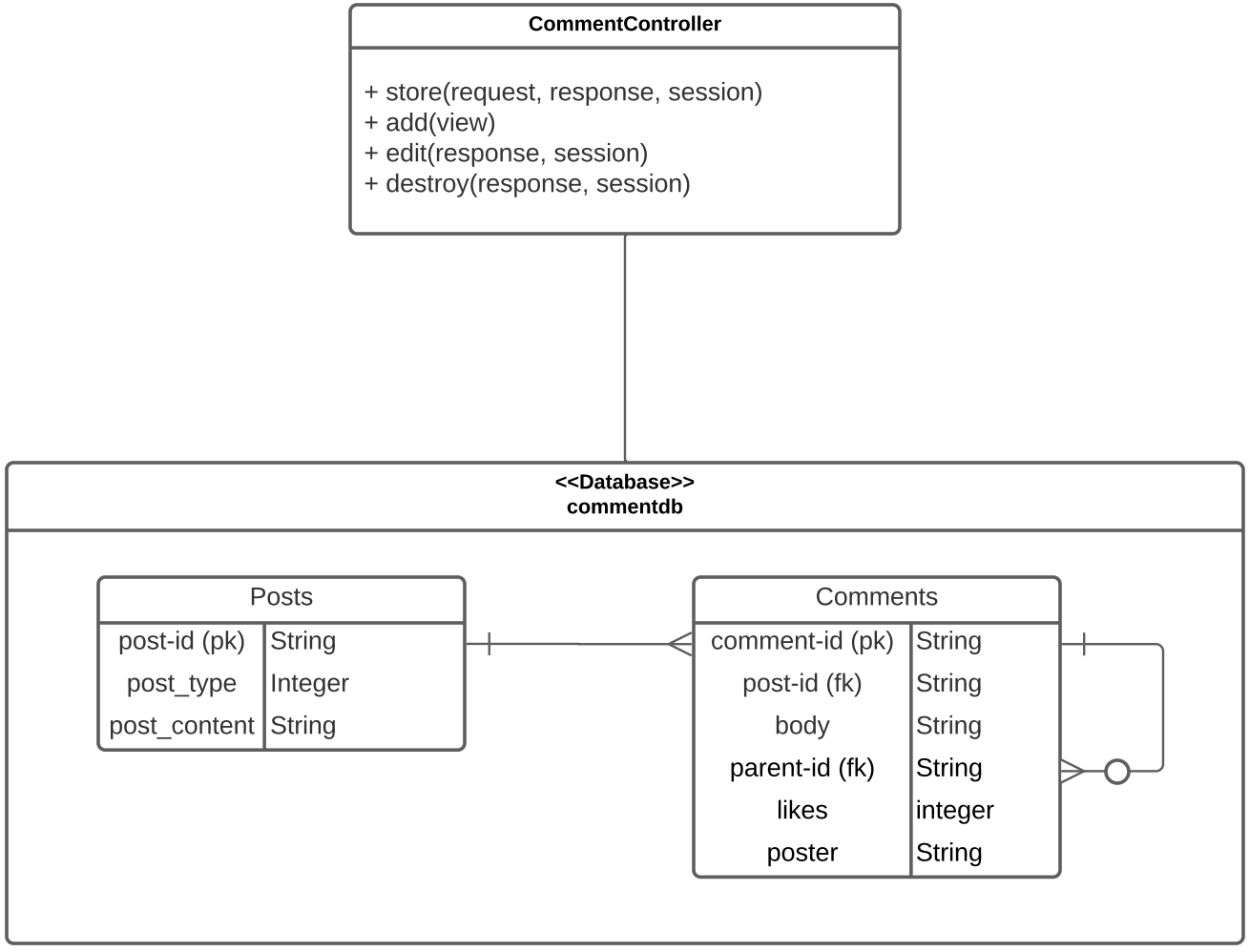

Design Decisions:

When designing our comment API, the biggest focus was on our database. Our databse is comprised of two tables, Posts and Comments. Each table has a generated primary key. We want our API to allow comments on images, videos, and text posts. This is why we have a post-type column in the Posts table. Because each post can have multiple comments, but each comment can only be on one post, there is a one to many relationship between posts and comments. As for the Comments, we wanted to allow comments on other comments. We agreed that "infinite subcomment" ie, a comment on a comment that's already a subcomment, can be hard to read and messy. Thus a comment can have zero or many subcomments, but subcomments can not have subcomments. Hence the one to zero or many relationship between comments. With our comment controller we wanted to be able to store comments to the database, add comments to a view, edit comments, and delete comments. The store method is responsible for taking a request, saving data to the database and sending a response. Add is responsible for putting a comment in a view. Edit and destroy each take a response and session as parameters and edit or delete a comment respectively.