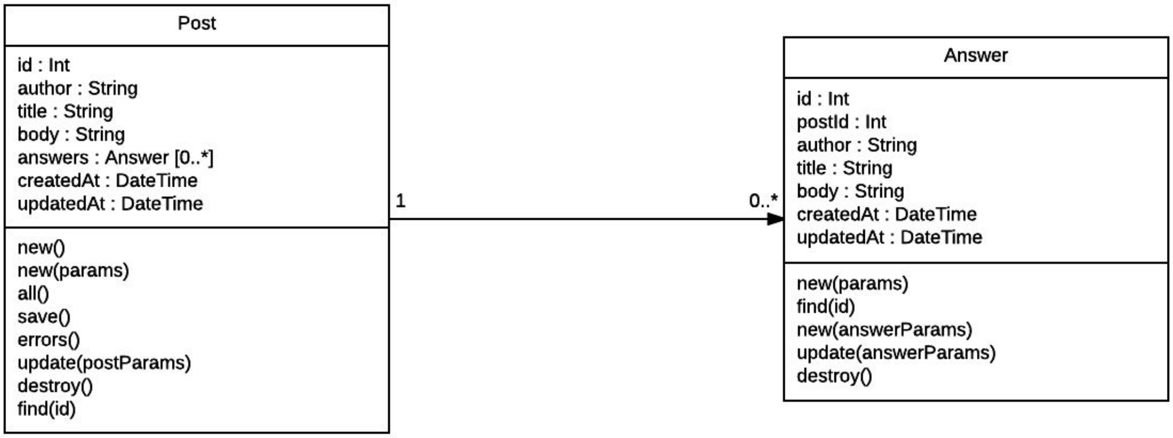
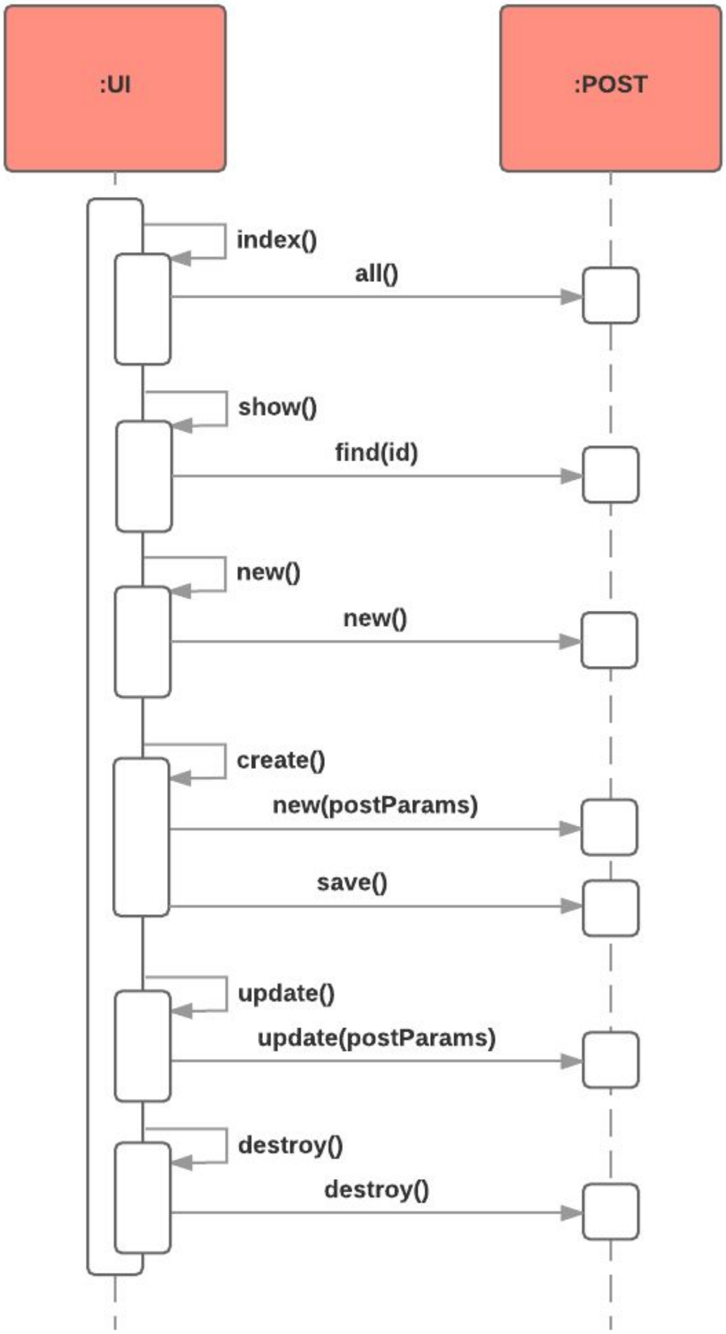
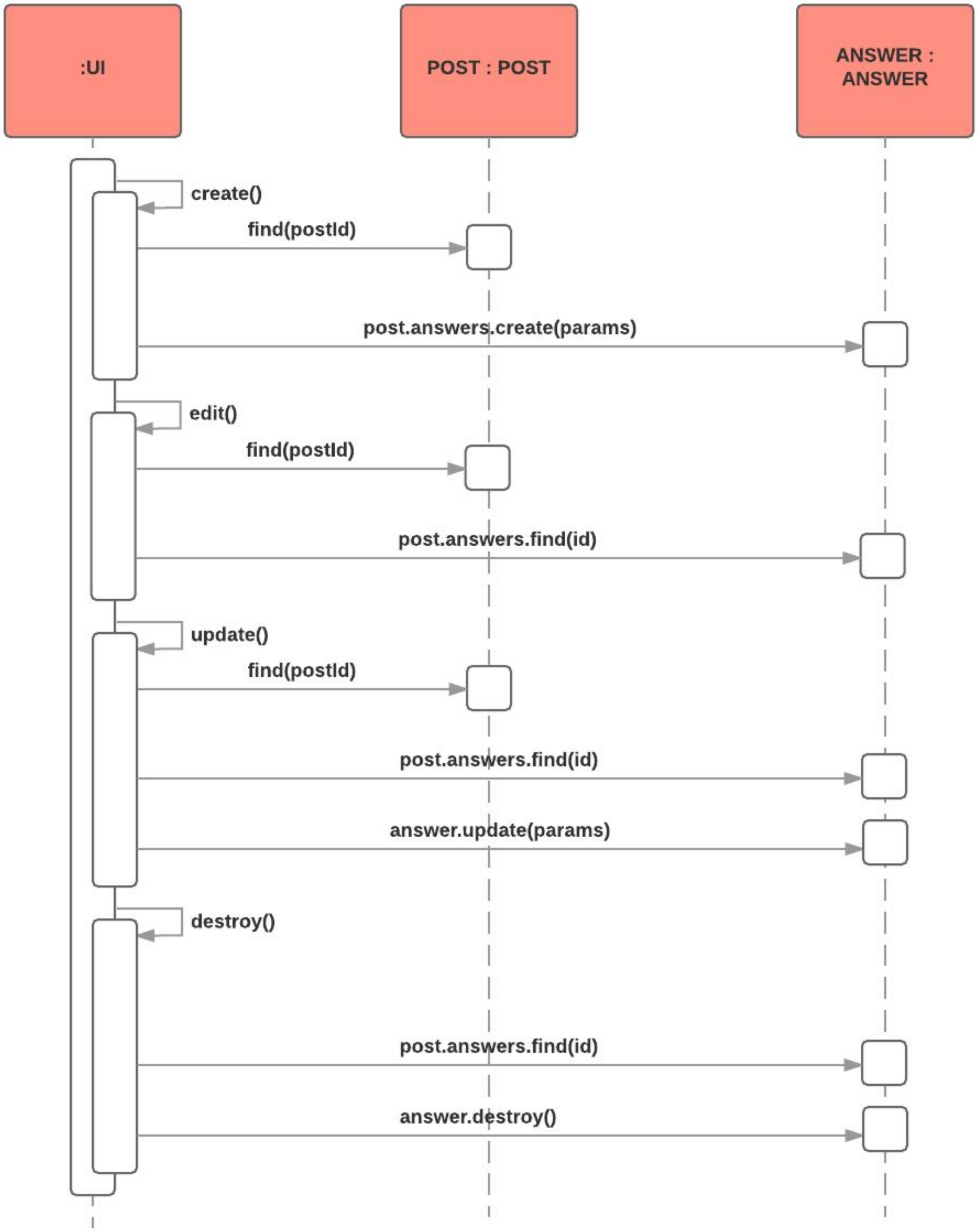
Table of Contents

I. Design Class Diagram With All Classes & Associations



II. Design Level Sequence Diagrams





III. Discussion of Design Decisions

When designing our application for iteration 1, we wanted to implement the iteration as simple as possible. With this in mind, we took full advantage of Rails' scaffolding capabilities in order to implement the business logic and flow of our application. However, we did not want to finish iteration 1 with a bare-bones implementation of the use cases we designated for this iteration. So, using the views and designs provided to us from Rails' scaffolding, we wanted to use those as the starting point to begin working out a full-fledged design of our application. We decided to take this direction so that we could provide a demo at the end of this iteration that is more representative of the vision we have for our application and the future directions we intend to take in future development.

IV. Discussion of Database Design

The database for iteration 1, did not need to be complex. In whole, the database consists of two tables (*Posts* and *Comments*). For the time being, each table has a small number of data variables. For *Posts*, each record in this table will have an *ID*, *author*, *title*, and *body* (in addition to the auto-generated *created_at* and *updated_at* data). For *Comments*, each record in this table will have an *ID*, *post_ID*, *author*, and *body* (in addition to the auto-generated *created_at* and *updated_at* data).

The relationship between these two tables is simple. A *post* can have many comments, and a *comment* must belong to a single *post*. This database was implemented with the following Rails commands and Ruby code below.

Rails Commands

```
$ bin/rails g scaffold Post title:string  body:text author:string
$ bin/rails g model Comment author:string body:text post:references
$ bin/rake db:migrate
```