

## Project Finders and Validations

### Class Project Overview

Your class project will consist of the design, development, testing and debugging of a Rails web application of your choice. Here you get to solve a problem of interest to you and you get to be creative. The project does not have to be anything fancy, but I do want it to involve some kind of database table. It can be most anything that includes Ruby and Rails. I hope you'll pick something that will be fun. There will be multiple assignments that deal with your class project as the semester progresses.

### Project Data Layer

At this time, I would like you to identify and describe which information you will allow to be searched and the kinds of data validations rules you plan to enforce. For example, if my class project were about my favorite songs and singers, I would create the following for this assignment:

Field Name	Description	Kind	Validation	Finder?
singerName	The person who sang the song	string	non-empty, length < 100 characters	yes
songTitle	The name of the song they sang	string	non-empty, length < 100 characters	yes
releaseYear	The year when the song got released	integer	non-empty, integral, value > 1950, value < 2014	no
albumTitle	The name of the album which first released this song	integer	non-empty, length < 100 characters	yes
favorite	Is this song really one of my favorites??	boolean	non-empty, boolean	yes
songLength	The time length of the song, in seconds	boolean	non-empty, integral, non-negative	no

The page here will give you the ability to enter this information right onto this assignment page below.

## Project Finders and Validations (FinanceApp)

USER				
Field Name	Description	Type	Validation	Finder?
ID (PK)		integer		
name	The user's username	string	non-empty; length < 25 characters; unique	no
password_digest	The user's password	string	non-empty; length > 6 characters	no
email	The user's email	string	non-empty; length < 25 characters; matches the official email standard	no
cash	The amount of cash	decimal	-	no
rails generate model user name:string password_digest:string email:string cash:decimal				
<b>\FinanceApp\app\models\user.rb</b>  <pre> class User &lt; ActiveRecord::Base   # A has_one association sets up a one-to-one connection with   # another model.   # Each user has only one portfolio.   has_one :portfolio    # A has_one :through association sets up a one-to-one connection   # with   # another model by proceeding through a third model. Each user   # has one   # portfolio, and each portfolio is associated with one portfolio   # history.   has_one :portfolio_history, :through =&gt; :portfolio end </pre>				

PORTFOLIO ( the collection of stocks that user own)				
Field Name	Description	Type	Validation	Finder?
ID (PK)		integer		
user_id (FK)	The user's ID.	integer	unique	no
symbol	The stock's symbol.	string	unique; valid (exists)	yes
shares	How many shares a user owns of a particular stock.	integer	non-negative integer	no
rails generate model portfolio user:references symbol:string shares:integer				
<b>\FinanceApp\app\models\portfolio.rb</b>  <pre> class Portfolio &lt; ActiveRecord::Base   # Sets up a one-to-one connection with another model.   # Each portfolio can be assigned to exactly one user. </pre>				

```

belongs_to :user

# A has_one association sets up a one-to-one connection with
# another model.
# Each portfolio has only one history (portfolio history).
has_one :portfolio_history
end

```

PORTFOLIO_HISTORY				
Field Name	Description	Type	Validation	Finder?
ID (PK)		integer		
portfolio_id (FK)	The portfolio's ID	integer	unique	no
transaction	Whether a stock was bought or sold	string	non-empty; "BUY" or "SELL"	no
date	The price of a share at the time of transaction.	date	valid	no
symbol	The symbol bought or sold.	string	valid (exists)	no
shares	The number of shares bought or sold.	integer	non-negative integer	no
price	The date and time of the transaction.	decimal	non-negative	no

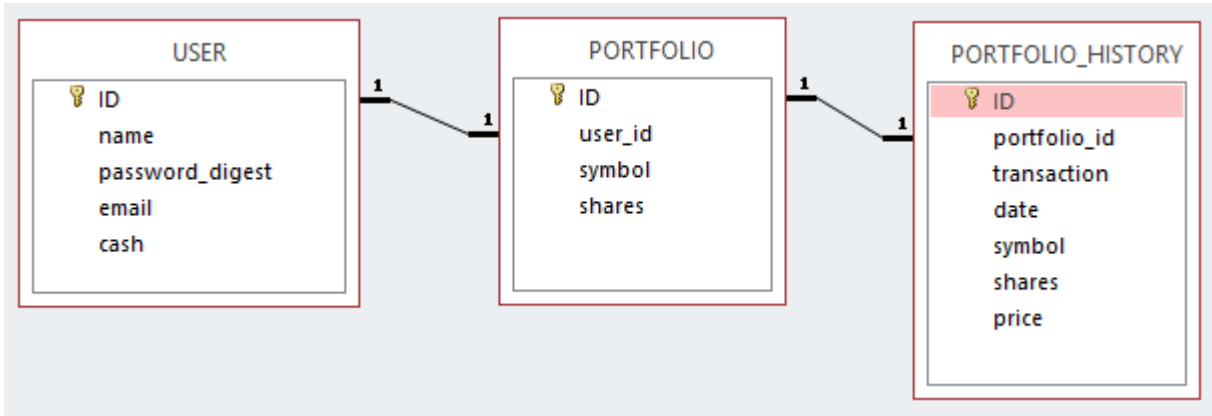
rails generate  
model portfolio\_history portfolio:references transaction:string  
date:timestamps symbol:string shares:integer price:decimal

**\FinanceApp\app\models\portfolio\_history.rb**

```

class PortfolioHistory < ActiveRecord::Base
  # Sets up a one-to-one connection with another model.
  # Each (portfolio) history can be assigned to exactly one
  # portfolio.
  belongs_to :portfolio_history
end

```



**Relational schema** (All relations are of ONE-TO-ONE type).