A **User**'s userId must be between 3 to 20 characters, unique, and cannot be optional. A **User**'s password must be between 6 to 8 characters, non-empty, and cannot be the same as the **User**'s userId. A **User** hasOne **Profile**, and it can be optional. A **User** can create many **Posts** and each **Post** belongs to only one **User**. A **User** can follow other **Users**.

A **Profile**'s only requirements are that it must have a fullName and an email. If a bio is supplied, the maxSize is 1000 characters long and if a photo is supplied, the maxSize is 2 MB.

A **Post** can never have content that is empty. **Posts** should be sorted in a descending order based on the dateCreated and they should also be sorted in a descending order when accessing **Posts** via the **User**. A **Post** has many **Tags**, and each **Tag** may relate to many **Posts**. As a result, a **User** creates many **Tags** and each **Tag** relates to one **User**. This means that a **Tag** belongsTo a **User** and a **Post**.

Check out the scaffolding!!

Replace TODO in **Post toString()** with a Groovy String that accesses a valid userId (HINT: requires the use of a groovy operator)

-------- MUST COMPLETE EVERYTHING BEFORE THIS BEFORE MOVING ON------

Write an error validation message in the *message.properties* file.

Rename main.gsp to oldmain.gsp. Fill in the missing GSP tags for toDoMain.gsp and rename it to main.gsp. Add the GSP tag that adds the custom css named "hubbub.css". Add the GSP tag that adds a custom logo located in "/images/headerlogo.png" Type in a custom footer.

Implement **createAdminUserIfRequired()** to *BootStrap.groovy* with whatever values you desire. Remember, you should not have duplicate admins.

Implement search for **Users.** Add the GSP tags in *search.gsp* to submit form and search by userId. You should have three GSP tags. Add search logic to *UserController*. (HINT: one of the methods do not need to be implemented). Fill in the TODO's in *results.gsp* with the corresponding Groovy strings returned from your **results()** function in *UserController*. Add the GSP tags that iterates through the users and shows a link to the users.

Make an advanced search for **Users** using criteria queries.

Add a timeline to **Posts** that gives users the ability to see all the **Posts** corresponding to that specific **User.** If you can't find the **User**, return error code 404. Fill in missing GSP tags for creating a form that allows users to add posts in *timeline.gsp*. Don't forget to implement addPost, so you will be able to link it to the form located in *timeline.gsp*

Add a homepage redirect, so that when users navigate to /bootcamp/post they are redirected to a timeline

Create a PostService with a method to **createPost()** and link it to PostController. Modify your **addPost()** method to use **postService.createPost()** (HINT: It will take two parameters)

Implement **register()** in *UserController*. Now do it again except with command object *UserRegistrationCommand* and implement the constraints and call it **newregister()**.

Implement functionality in *ImageController* to handle image uploading and rendering. Fill in the missing GSP tags for upload and selection in */image/form.gsp*. Implement corresponding *profile* action in *UserController*

Implement *LameSecurityFilters* and corresponding *LoginController* with methods **form()** and **signIn(String userId, String password)**

Add the following URL Mappings, so that when you access a userId, it redirects to their timeline

Add the ability to show which friends you are following in UtilTagLib.

Implement the global timeline.

Rename "toDoTimeline.gsp" to timeline.gsp and rename timeline.gsp to oldtimeline.gsp

Implement addPostAjax for PostController.