Project 2 notes
ToDo List App
Habitu.al

Used API's:
Email - Some Email API
Quote of the day - Random Famous Quotes API
Activity suggestions - Bored

User stories:

User can create account
Ask for a password, email address4(username)
Insert into the database
If email exists exception: "Email already exists"

*userRepo.exists(User class) or *catch alreadyExists

If not, insert email address and (encoded)password

userRepo.register(User class)

(Optional) verify email
Create and save some token attributed to a user id
Send email link to user "example.com/verify/<token>"
If they click it, verify token exists and verify user if true

User can log in
User inputs email & password;
Check if email exists in db;
If email not exists: "Invalid Login"
If email exists: check if (encoded)password matches password in db
If doesn't match: "Invalid Login"
If exists construct User;
Save User as currentUser in session;

User can add tasks to their profile

Click (+) button

- input name
- Is Single or Routine Task?
    - If Single:
        - Input optional completeByDate, description, remind_by
    - If Routine:
        - Input required completeByDate, repeats_every value
        - Input optional description, remind by
- If Group task, create task for everyone in the group

- Insert into DB newTask

User can resolve tasks

User clicks checkbox

- Update that task isDone = true
- If singletask: change category to "Complete"

User can create category

* "Complete" is not a valid category name

User clicks add category

- Input name *
- Validate that category is not complete -> "Invalid Category Name"
- Insert into category DB -> name, userId

User can view tasks ordered by category

User clicks "view tasks"

- Get all tasks by user id from DB
- Sort into categories
- Render

User can put tasks into categories

User drags task into category “bucket”

- Update task.category in task db

User can create a group

User clicks create group

- Input group name
- Insert into group DB -> name;
- Assign this user to this group

User can view their groups

User clicks “view groups”

- Get all groups by user id from DB
- Render

User can assign people to group

User clicks add people button

- Input email address
- Validate email in the DB
    - If not in db “User not found”
- Insert email address into group member DB
- Add groups tasks to that users tasks

User (admin) can assign people as admin in group

User selects a person

- Update that group member isAdmin to True

taskRepo.resolve(task)

If user is bored and wants a random activity click this button!

User clicks “bored” button

- Run the bored api

User can share app on twitter and facebook

User clicks “bored” button

- Run the bored api

TBD REMINDER FEATURE

User can set up reminder for their task
User selects task and edits reminder

User can “snooze” the reminder for X time.
Snooze will just be adding to (or resetting) the reminder

MODELS

PUBLIC ABSTRACT CLASS task	Public class OneTimeTask extends task	Public class RoutineTask extends task	Public class categories
String name, String desc	String message, int timeToExpire	String message, int timeToExpire, int timeBetweenTask	String type, Object routineTask, Object OneTimeTask
Public class User			
String name, String password, String email,			

CONTROLLERS

DAO’s
