two web-based interfaces: ***userUI*** and ***managerUI***.

UserUI: 第一个作业中的

***managerUI*** should support the following functionality:

1. List workers, and their CPU utilization
2. Manually growing the worker pool by 1.
3. Manually shrinking the worker pool by 1.
4. Functionality for configuring a simple auto-scaling policy by setting the following parameters:
   * CPU threshold for growing the worker pool
   * CPU threshold for shrinking the worker pool
   * Ratio by which to expand the worker pool (e.g., a ratio of 2 doubles the number of workers).
   * Ratio by which to shrink the worker pool (e.g., a ratio of 4 shuts down 75% of the current workers).
5. Functionality for deleting all data. Executing this function should delete application data stored on the database as well as all images stored on S3.

**Requirements**

1. All images should be stored in S3
2. Run only 1 instance of the database (i.e., do not replicate the database); all workers should communicate with this single database instance.
3. Use a separate EC2 instance to run each worker node. All worker nodes should run the ***userUI*** functionality.
4. The ***managerUI*** should run on a single EC2 instance separate from the instances used to run the worker nodes.
5. Your application should monitor the load on its pool of workers using the AWS CloudWatch API. When the load exceeds (or drops bellow) the configurable threshold, your application should use the AWS EC2 API to resize its worker pool. Do **NOT** use AWS Auto Scaling feature for this assignment. ***Important***: in order to use AWS CloudWatch, you need to enable monitoring of your EC2 instances.
6. Use only EC2 t2.small instances for your worker pool

Use EC2 Load Balancing to distribute request among your workers. Your application should use the AWS Elastic Load Balancing API to add and remove workers from the load balancing pool as you reconfigure your worker pool in response to load. (“Classic Load Balancer.”)

1. A load generator that conforms to this interfaces is available [here](http://www.cs.toronto.edu/~delara/courses/ece1779/projects/ece1779LoadGenerator.tar). **Warning:** the load generator creates a lot of network traffic. To minimize bandwidth charges, please use the load generator inside EC2 only.

To run the program untar the file, and cd into ece1779LoadGenerator/bin

Run as: java -cp . ece1779.loadgenerator.LoadGenerator server\_ip\_address port userid password

1. UserUI instance belong to one of the worker node, which means the default number of worker is 1