Designing a pid Manager and its test program

You must use 'Java' language.

1. Before going to the main problem. You first need to implement a test program with which you can test the validity of the pid Manager you will implement. The program simply takes the number of threads created and the life time of the program (i.e. how many seconds the program will be running). It also takes the life time of a thread while the program is running. While the program is running, threads are created in a random time and simply doing nothing until it will be destroyed. The output of the program will be as follows:

   ```
   Test program is initialized with 10 thread and 60 seconds, with the life time 10 seconds of each thread
   Thread 1 created at Second 10
   Thread 2 created at Second 15
   Thread 1 destroyed at Second 20
   Thread 3 created at Second 21
   ....
   60 seconds has passed... Program ends
   ```

2. Implement the pid Manager. Please be careful in developing getPIDWait(). Also, the class providing the interface PIDManager may take MAX_PID and MIN_PID in order to control the range of IDs used.

3. Test your pid Manager implementation by modifying the program you made in step 2. Both getPID() and getPIDWait() should be tested. You need to provide how you are able to test your pid Manager with the test program.

Step	**Results**	**Points**
Program in Step 1	Program is working successfully following the intention in the assignment	15
Program in Step 2	You could develop a class implementing PIDManager interface	10
Program in Step 2 (2)	getPID() is working well	10
Program in Step 2 (3)	getPIDWait() works well	30
Program in Step 3	The testing program is working well and we can test both getPID() and getPIDWait() with the program.	15
Error handling		10
documentation		10