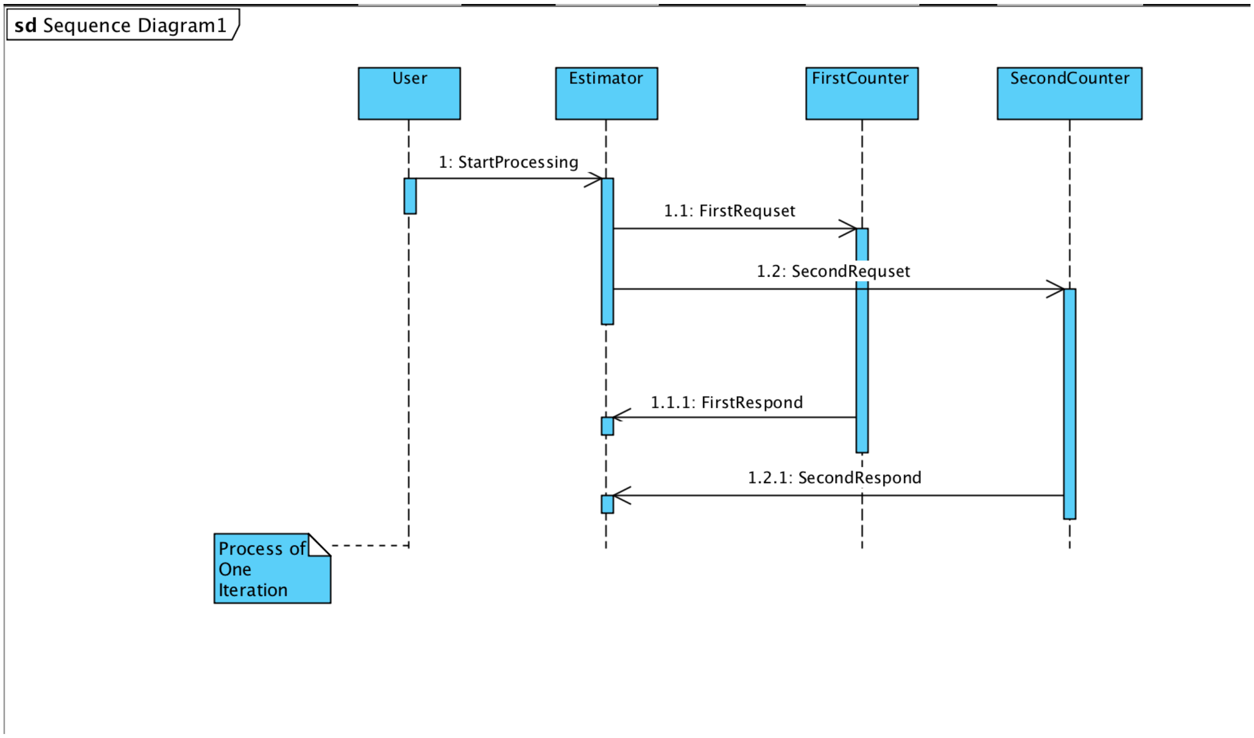
CSYE7215 Parallel & Multithreaded Prog
Homework 9
Sequence Diagram and Pseudo Code
Wenhe Ma

Sequence Diagram:



Pseudo Code:

Main:

```
public class User {
	public static void main(String[] args) throws Exception {
		//CREATE the ActorSystem
		//CREATE the actor called estimator
		//GET an array of File called files from the fold
		//FOR EACH File in the array of File called files
			//SEND estimator the StartProcessingFolderMessage
		//ENDFOR
		//SHUTDOWN the ActorSystem
	}
	public static File[] readDirectory() {
		//CREATE File with path to the folder
		//GET files from folder
		//RETURN listOfFiles
	}
}
```

Actor classes:

```
public class Estimator extends UntypedActor {
```

```java
//CREATE Props object for Estimator class
//INIT global variable double PARAMETER_P2 to 0.9
//INIT global variable double PARAMETER_P3 to 1.1
//INIT global variable String text to null
//INIT global variable int result1 to 0
//INIT global variable int result2 to 0
//INIT global variable int estimatedResult to 0
//INIT global variable int estimatedResult to 0
//INIT global variable boolean receivedFirstMessage to false
//INIT global variable boolean receivedSecondMessage to false

@Override
public void onReceive(Object msg) throws Throwable {
    //IF msg IS INSTANCE OF StartProcessingFolderMessage
        //GET file from msg
        //READ the text file and SET text
        //DIVIDED the text file
        //CREATE FirstCounter
        //CREATE FirstRequestMessage with 1st part of text
        //SEND FirstRequestMessage to firstCounter
        //CREATE SecondCounter
        //CREATE SecondRequestMessage with 2nd part of text
        //SEND SecondRequestMessage to secondCounter
    //ELSE IF msg IS INSTANCE OF FirstRespondMessage
        //GET result1 from msg
        //SET estimatedResult TO MULTIPLY result1 BY 2 and by P1
        //PRINT result1
        //SET receivedFirstMessage TO true
        //IF estimator received the SecondMessage
            //CALCULATE the trueResult
            //PRINT the trueResult
            //PRINT the estimatedResult
            //PRINT P1
            //IF trueResult IS GREATER THAN estimatedResult
                //INCRE P1 BY MULTIPLY it by P3(1.1)
            //ELSE IF trueResult IS LESS THAN estimatedResult
                //DECRE P1 BY MULTIPLY it by P2(0.9)
            //ENDIF
            //PRINT fixed P1
        //ENDIF
    //ELSE IF msg IS INSTANCE OF SecondRespondMessage
        //GET result2 from msg
        //PRINT result2
        //SET receivedSecondMessage TO true
        //IF estimator received the FirstMessage
            //CALCULATE the trueResult
            //PRINT the trueResult
            //PRINT the estimatedResult
            //PRINT P1
            //IF trueResult IS GREATER THAN estimatedResult
                //INCRE P1 BY MULTIPLY it by P3(1.1)
            //ELSE IF trueResult IS LESS THAN estimatedResult
                //DECRE P1 BY MULTIPLY it by P2(0.9)
            //ENDIF
            //PRINT fixed P1
        //ENDIF
    //ELSE
```

```
            //PRINT error message
            //unhandled msg
        //ENDIF
    }
}
```

In my implementation, First Counter and Second Counter are doing the exact same job. I decided to write separate classes for First and Second Counter just in order to make it easier to distinguish the 2 counters.

```
public class FirstCounter extends UntypedActor {
    //CREATE Props object for FirstCounter class
    @Override
    public void onReceive(Object msg) throws Throwable {
        //IF msg IS INSTANCE OF FirstRequestMessage
            //GET text from msg
            //COUNT vowels in the text, SET it to result1
            //CREATE FirstRespondMessage with PARAMETER result1
            //SEND FirstRespondMessage back to Estimator
        //ELSE
            //PRINT error message
            //unhandled msg
        //ENDIF
    }
    //INIT List of Character called vowelList with 'A', 'E', 'I', 'O','U','Y',
'a','e','i','o','u', 'y'
    private int vowelCount(String input) {
        //INIT count to 0
        //FOR EACH char in the input String
            //IF vowelList CONTAINS the char
                //INCRE count by 1
          //ENDIF
        //END LOOP
        //RETURN count
    }
}
```

Message classes:

```
public class StartProcessingFolderMessage {
    //DEFINE private final variable file, the message passed from User to Estimator
    //DEFINE Constructor of StartProcessingFolderMessage with field file
    //DEFINE GETTER of field file
}
```

Again, First messages and Second messages are the same thing.

```
public class FirstRequestMessage {
    //DEFINE private final variable text, the message passed from Estimator to Counter
    //DEFINE Constructor of FirstRequestMessage with field text
    //DEFINE GETTER of field text
}
public class FirstRespondMessage {
    //DEFINE private final Integer result, the message passed from Counter to Estimator
    //DEFINE Constructor of FirstRespondMessage with field result
    //DEFINE GETTER of field text
}
```