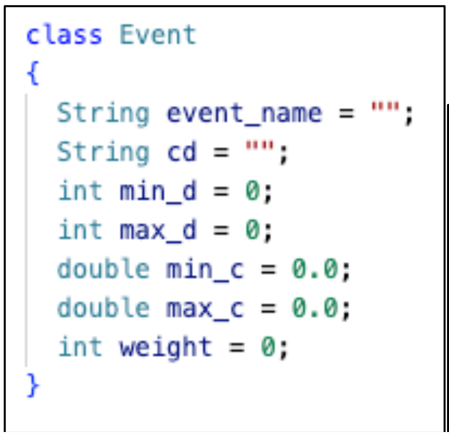
CSCI262-A3
Ang Kian Guan
7023224
T06P

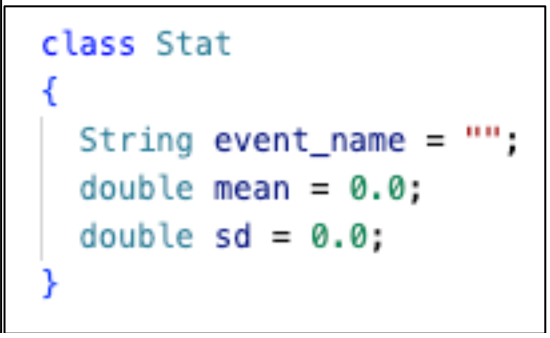
Initial Input

1. To store Events and Stats input from the txt files, the progam uses Event and Stat Objects to store the data collected. The Input class is responsible to read the text files, and store each line entry into the object before adding the objects to the Events and Stats Arraylist respectively.

Fig 1. Classes to store each event and stat as objects

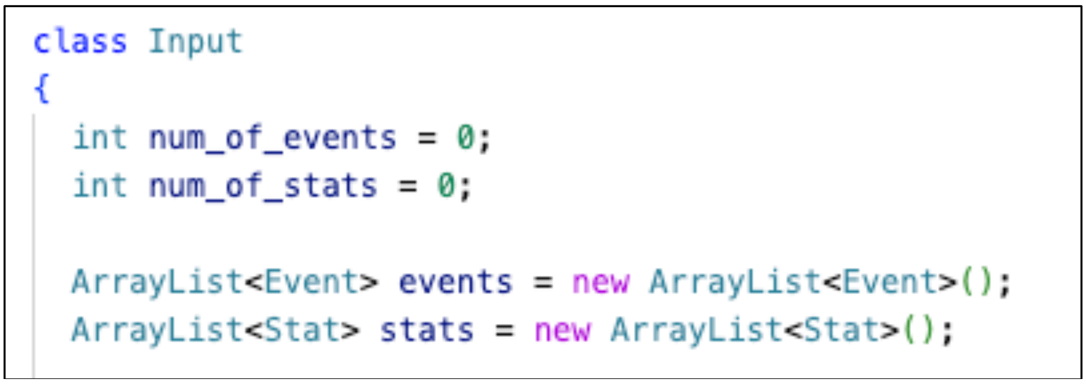


```
class Event
{
  String event_name = "";
  String cd = "";
  int min_d = 0;
  int max_d = 0;
  double min_c = 0.0;
  double max_c = 0.0;
  int weight = 0;
}
```



```
class Stat
{
  String event_name = "";
  double mean = 0.0;
  double sd = 0.0;
}
```

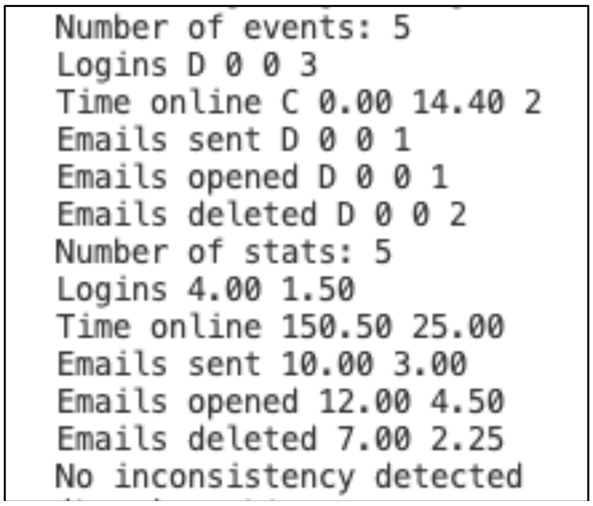
Fig 2. Input Class to read and store events and stats in an Arraylist



```
class Input
{
  int num_of_events = 0;
  int num_of_stats = 0;

  ArrayList<Event> events = new ArrayList<Event>();
  ArrayList<Stat> stats = new ArrayList<Stat>();
```

2. To check if there are inconsistencies between the txt files, a consistency check function is implemented to compare event names in the two files. Event names are captured in separate sorted treeset and compared to identify discrepancies. The program highlights any discrepancies but does not go into details.

Fig 3. Output of Initial Input

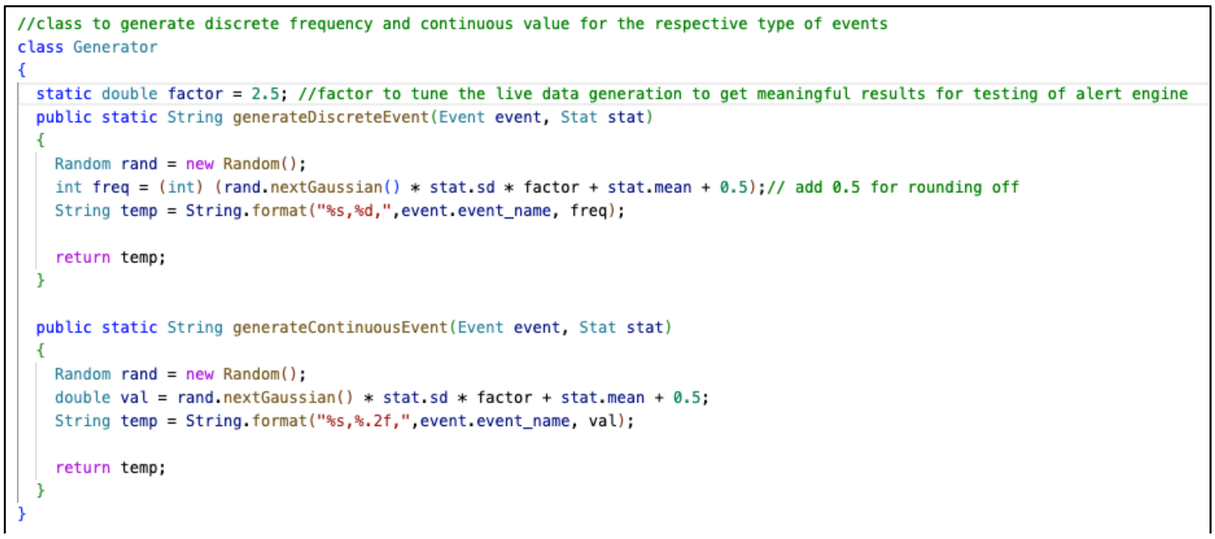


```
Number of events: 5
Logins D 0 0 3
Time online C 0.00 14.40 2
Emails sent D 0 0 1
Emails opened D 0 0 1
Emails deleted D 0 0 2
Number of stats: 5
Logins 4.00 1.50
Time online 150.50 25.00
Emails sent 10.00 3.00
Emails opened 12.00 4.50
Emails deleted 7.00 2.25
No inconsistency detected
```

Activity Simulation Engine and Logs

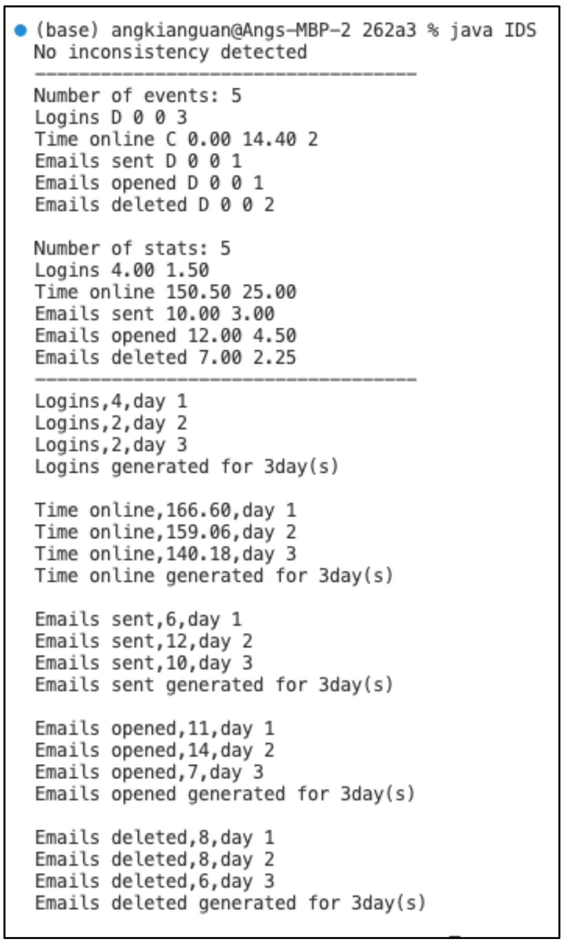
1. The gaussian() function in the java Random Class was used to generate the events. We are able to vary the standard deviation of this output by multiplying it with a factor. This proved to be very useful for the testing of alert engine later in the program.

Fig 4. Generate values according to normal distribution



```java
//class to generate discrete frequency and continuous value for the respective type of events
class Generator
{
  static double factor = 2.5; //factor to tune the live data generation to get meaningful results for testing of alert engine
  public static String generateDiscreteEvent(Event event, Stat stat)
  {
    Random rand = new Random();
    int freq = (int) (rand.nextGaussian() * stat.sd * factor + stat.mean + 0.5);// add 0.5 for rounding off
    String temp = String.format("%s,%d,",event.event_name, freq);

    return temp;
  }

  public static String generateContinuousEvent(Event event, Stat stat)
  {
    Random rand = new Random();
    double val = rand.nextGaussian() * stat.sd * factor + stat.mean + 0.5;
    String temp = String.format("%s,%.2f,",event.event_name, val);

    return temp;
  }
}
```

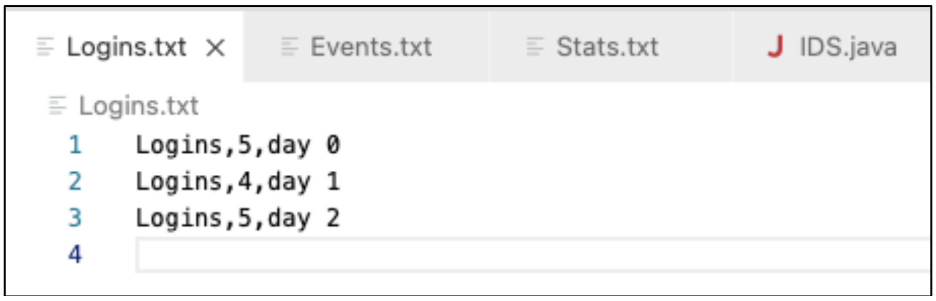
Fig 5. Example of values generated vs statistics



```
(base) angkianguan@Angs-MBP-2 262a3 % java IDS
No inconsistency detected
-----------------------------------
Number of events: 5
Logins D 0 0 3
Time online C 0.00 14.40 2
Emails sent D 0 0 1
Emails opened D 0 0 1
Emails deleted D 0 0 2

Number of stats: 5
Logins 4.00 1.50
Time online 150.50 25.00
Emails sent 10.00 3.00
Emails opened 12.00 4.50
Emails deleted 7.00 2.25
-----------------------------------
Logins,4,day 1
Logins,2,day 2
Logins,2,day 3
Logins generated for 3day(s)

Time online,166.60,day 1
Time online,159.06,day 2
Time online,140.18,day 3
Time online generated for 3day(s)

Emails sent,6,day 1
Emails sent,12,day 2
Emails sent,10,day 3
Emails sent generated for 3day(s)

Emails opened,11,day 1
Emails opened,14,day 2
Emails opened,7,day 3
Emails opened generated for 3day(s)

Emails deleted,8,day 1
Emails deleted,8,day 2
Emails deleted,6,day 3
Emails deleted generated for 3day(s)
```

2. To organise the logs, the files are named after their respective event. Within the file, each line captures the event name, the frequency and the day it was recorded. The values are separated by commas. This will allow the data to be split and used for processing at a later stage.

Fig 5. Sample log file content

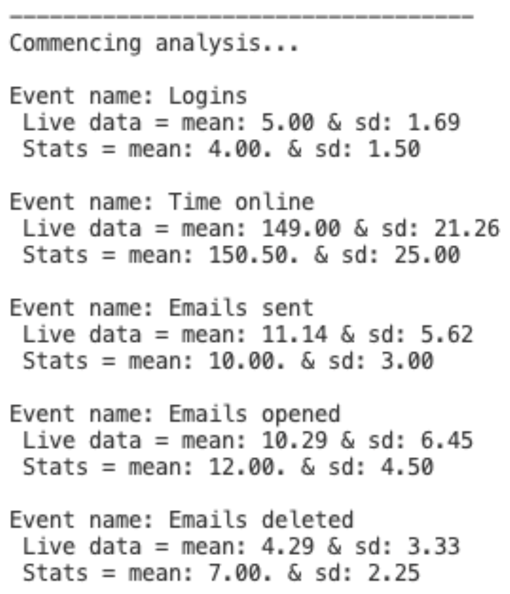


```
Logins,5,day 0
Logins,4,day 1
Logins,5,day 2
```

Analysis Engine and Log File

1. During analysis, the program will display each event's statistics and compare it against the provided statistics:

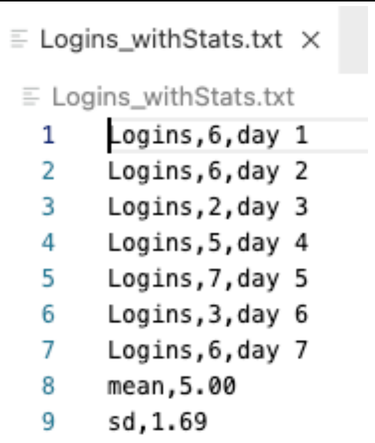
Fig 6. Sample analysis engine runtime report



```
-----------------------------------
Commencing analysis...

Event name: Logins
 Live data = mean: 5.00 & sd: 1.69
 Stats = mean: 4.00. & sd: 1.50

Event name: Time online
 Live data = mean: 149.00 & sd: 21.26
 Stats = mean: 150.50. & sd: 25.00

Event name: Emails sent
 Live data = mean: 11.14 & sd: 5.62
 Stats = mean: 10.00. & sd: 3.00

Event name: Emails opened
 Live data = mean: 10.29 & sd: 6.45
 Stats = mean: 12.00. & sd: 4.50

Event name: Emails deleted
 Live data = mean: 4.29 & sd: 3.33
 Stats = mean: 7.00. & sd: 2.25
```

2. The file generated is named "<event name>_withStats.txt". In the file, the data format is <event name><frequency><day number> followed by mean and standard deviation.

Fig 7. Sample file of events and statistics

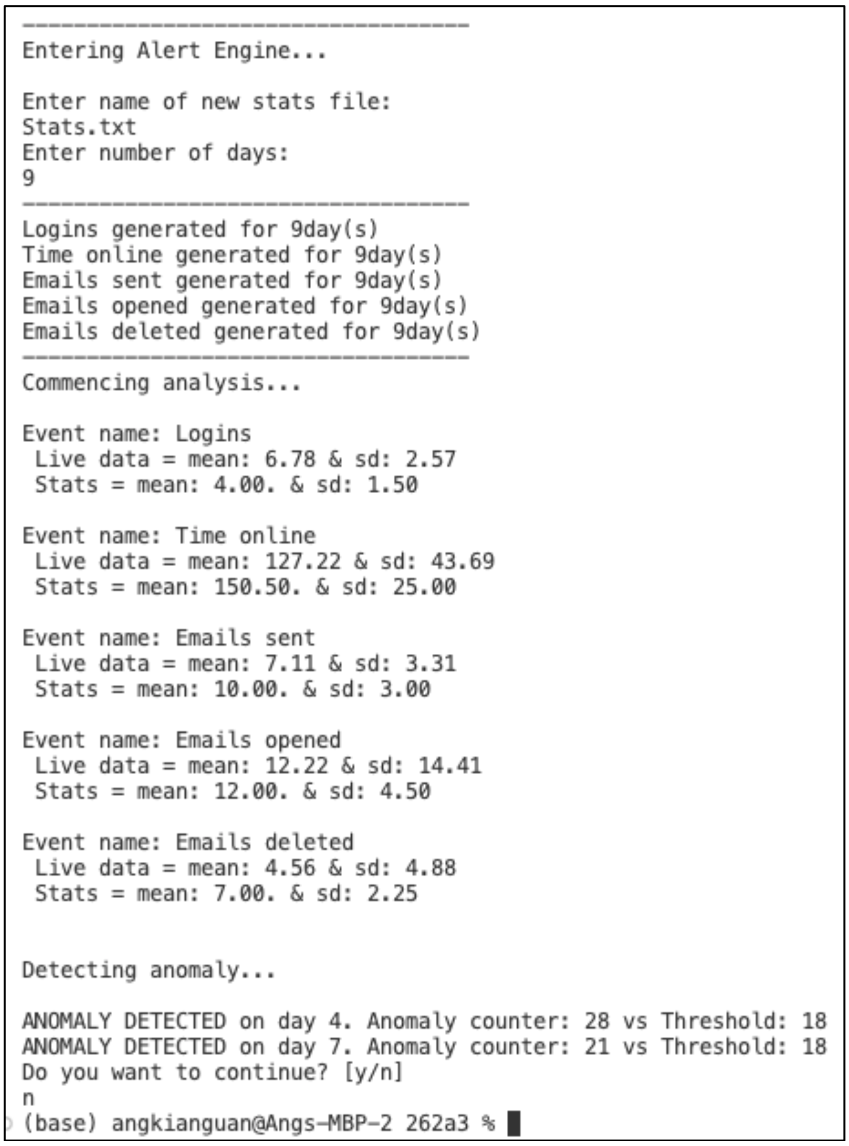


```
Logins,6,day 1
Logins,6,day 2
Logins,2,day 3
Logins,5,day 4
Logins,7,day 5
Logins,3,day 6
Logins,6,day 7
mean,5.00
sd,1.69
```

Alert Engine

1. The output from the alert engine:

Fig 8. Sample output of alert engine



```
-----------------------------------
Entering Alert Engine...

Enter name of new stats file:
Stats.txt
Enter number of days:
9
-----------------------------------
Logins generated for 9day(s)
Time online generated for 9day(s)
Emails sent generated for 9day(s)
Emails opened generated for 9day(s)
Emails deleted generated for 9day(s)
-----------------------------------
Commencing analysis...

Event name: Logins
 Live data = mean: 6.78 & sd: 2.57
 Stats = mean: 4.00. & sd: 1.50

Event name: Time online
 Live data = mean: 127.22 & sd: 43.69
 Stats = mean: 150.50. & sd: 25.00

Event name: Emails sent
 Live data = mean: 7.11 & sd: 3.31
 Stats = mean: 10.00. & sd: 3.00

Event name: Emails opened
 Live data = mean: 12.22 & sd: 14.41
 Stats = mean: 12.00. & sd: 4.50

Event name: Emails deleted
 Live data = mean: 4.56 & sd: 4.88
 Stats = mean: 7.00. & sd: 2.25


Detecting anomaly...

ANOMALY DETECTED on day 4. Anomaly counter: 28 vs Threshold: 18
ANOMALY DETECTED on day 7. Anomaly counter: 21 vs Threshold: 18
Do you want to continue? [y/n]
n
(base) angkianguan@Angs-MBP-2 262a3 % █
```

Disclaimer:

The program does not check the validity of user inputs and does not conduct error handling in such cases.