Preliminary:

1. Have ANSI-Color enabled terminal installed on IoT computer (Ubuntu, Cygwin, etc.)
2. Install latest Java JDK which will require Internet access or local installation storage

Setup:

All .class files should be in the same directory from which the program will be launched
As well should be included these files

users.hrt - lists all users that are authorized to perform actions on the selected train.
Format:= {username} {password} {role: cond/tech} {Preferred Name}

sensors.hrt - lists all sensors physically installed onboard the selected train.
System will not be able to communicate with sensors that are not mentioned in this file.
Format := {type: GPSS/DRS/IFS/CGS/WS} {Sensor_ID} {Train_Car_ID} {data pertaining to the sensor} ...
GPS S. := GPSS {ID} {Car_ID}
DR S. := DRS {ID} {Car_ID} {Position: 1/2} {Outgoing Frequency}
IF S. := GPSS {ID} {Car_ID}
CG S. := CGS {ID} {Car_ID}
W S. := WS {ID} {Car_ID} {Axel} {Wheel Diameter (in.)}

log.hrt - if none detected on startup, one will be made. Logs all events occuring in the IoT
Format:= {day/month/year} {hour:minute:second} {{event source}} {event data}
Event source can either be:

1. A username to represent actions explicitly done by a username
2. IOTSYS - to represent actions automatically done by system like connecting to sensors
3. GNRS - to represent generic updates that happen every router cycle
4. INFO - to represent an inform signal being display to the conductor
5. ALERT - to represent an alert signal being displayed to the conductor

testscript.hrt (for testing) - in order to simulate a real train, the IoT will read from this file to supply sensor data.
Format := {Sensor_ID} {data pertaining to the sensor} ...
GPS S. := {ID} {Latitude (deg.)} {Longitude (deg.)} {Time since last update (hrs.)}
DR S. := {ID} {Return frequency (Hz)} {Pulse duration (sec.)}
IF S. := {ID} {Temperature (C)} {Humidity (%)}
CG S. := {ID} {Light detected: 1/-1} {Sound detected: 1/-1} {Distance to gate: (mi.)/-1}
W S. := {ID} {RPM}

Usage:

1. Extract all files into the same directory
2. Start program with "$java IoTSimulator"
3. Log in to the system with provided credentials

If Conductor Log in:

   1. Confirm that no errors are produced on identification of sensors onboard
   2. When ready to set off, confirm the start of trip
   3. System will automatically log conductor out when no data is being supplied from sensors

If Technician Log in:

   1. Select option from list of options by typing its corresponding command [$command]
   2. When done, use logout

4. To quit the program, type 'quit' in the username field and hit enter