1.Example of cheating types to be detected :

These are outlined at the data/condition*.gz files.

Condition 1: With given TransitCard, the hash value does not match.
When hashed with HashUtil.hash(TransitCard), if the hash does not match, this is a violation.

```
{ "station": "vega","passenger": {"id": "18d8832c-6e83-41f9-84a0-5d31d9a2ff8a","florples": 88,
"history": [{"date": "2019-07-01T00:40:56.639945Z","change": -3, "station": "vega"},
{"date": "2019-06-24T10:05:13.639945Z","change": -3,"station": "m87"},
{"date": "2019-06-19T13:54:28.639945Z","change": -3,"station": "m87"},
{"date": "2019-06-14T10:43:16.639945Z","change": -3,"station": "pluto"},
{"date": "2019-06-11T22:51:24.639945Z","change": 100,"station": "pluto"} ],
"hash": "tZDQQHFOJ0LBRmsJkQfkJxshy3/OKV5EqqWJonUkZ8Q="}}
```

Condition 2: If the previous record for the same ID’s transaction history is greater or equal to the current ID, We know that this is duplicated.

Below, two different records coming from same ID have same length of the transaction history. Note that the first record departed at 2019-07-01 **00**:40:56 whereas second one departed at 2019-07-01 **02**:40:56

```
{"station":"vega","passenger":{"id":"18d8832c-6e83-41f9-84a0-5d31d9a2ff8a","florples":88,
"history":[{"date":"2019-07-01T00:40:56.639945Z","change":-3,"station":"vega"},
{"date":"2019-06-24T10:05:13.639945Z","change":-3,"station":"m87"},
{"date":"2019-06-19T13:54:28.639945Z","change":-3,"station":"m87"},
{"date":"2019-06-14T10:43:16.639945Z","change":-3,"station":"pluto"},
{"date":"2019-06-11T22:51:24.639945Z","change":100,"station":"pluto"}],
"hash":"uZDQQHFOJ0LBRmsJkQfkJxshy3/OKV5EqqWJonUkZ8Q\u003d"}}

{"station":"vega","passenger":
{"id":"18d8832c-6e83-41f9-84a0-5d31d9a2ff8a","florples":88,
"history":[{"date":"2019-07-01T02:40:56.639945Z","change":-3,"station":"vega"},
{"date":"2019-06-24T10:05:13.639945Z","change":-3,"station":"m87"},
{"date":"2019-06-19T13:54:28.639945Z","change":-3,"station":"m87"},
{"date":"2019-06-14T10:43:16.639945Z","change":-3,"station":"pluto"},
{"date":"2019-06-11T22:51:24.639945Z","change":100,"station":"pluto"}],
"hash":"c8UGJtT8+5E9OJXOifU7dcx3OKSWWw6NBN0yTIvt3yk\u003d"}}
```

Condition 3-1: If the transit card is duplicated before the previous departure, transaction records of the previous departure and the current departure after the duplication will be different.

```
{"station":"vega","passenger":{"id":"bda162ae-f1d3-48d0-a031-87cec14cf2e9","florples":85,
"history":
[{"date":"2019-07-17T09:32:57.460892Z","change":-3,"station":"vega"},
{"date":"2019-07-13T04:59:15.460892Z","change":-3,"station":"m87"},
{"date":"2019-07-09T02:54:15.460892Z","change":6,"station":"m87"},
{"date":"2019-07-03T18:33:42.460892Z","change":-3,"station":"centaurus"},
{"date":"2019-06-28T16:10:02.460892Z","change":-3,"station":"vega"},
{"date":"2019-06-25T02:22:14.460892Z","change":-3,"station":"m87"},
{"date":"2019-06-21T12:43:28.460892Z","change":-3,"station":"m87"},
{"date":"2019-06-15T20:36:50.460892Z","change":-3,"station":"betelgeuse"},
{"date":"2019-06-11T22:51:24.460892Z","change":100,"station":"betelgeuse"}],
"hash":"XWb+os58uVuoUQrqi8My9EuuPO525c1ThhnV0bW4M5w\u003d"}}

{"station":"vega","passenger":{"id":"bda162ae-f1d3-48d0-a031-87cec14cf2e9","florples":73,
"history":
[{"date":"2019-07-31T13:28:07.460892Z","change":-3,"station":"vega"},
{"date":"2019-07-26T21:18:18.460892Z","change":-3,"station":"betelgeuse"},
{"date":"2019-07-22T19:33:28.460892Z","change":-3,"station":"betelgeuse"},
{"date":"2019-07-17T14:53:18.460892Z","change":-3,"station":"betelgeuse"},
{"date":"2019-07-13T22:49:50.460892Z","change":-3,"station":"betelgeuse"},
{"date":"2019-07-09T02:59:15.460892Z","change":-3,"station":"m87"},
{"date":"2019-07-09T02:54:15.460892Z","change":6,"station":"m87"},
{"date":"2019-07-03T18:33:42.460892Z","change":-3,"station":"centaurus"},
{"date":"2019-06-28T16:10:02.460892Z","change":-3,"station":"vega"},
{"date":"2019-06-25T02:22:14.460892Z","change":-3,"station":"m87"},
{"date":"2019-06-21T12:43:28.460892Z","change":-3,"station":"m87"},
{"date":"2019-06-15T20:36:50.460892Z","change":-3,"station":"betelgeuse"},
{"date":"2019-06-11T22:51:24.460892Z","change":100,"station":"betelgeuse"}],
"hash":"JHpFS02zOBXobAJAz0kIB8xI1FcDGkHfhZ2qJn4kaOs\u003d"}}
```

From above, we can see that after the following record :
{"date":"2019-07-09T02:54:15.460892Z","change":6,"station":"m87"}
the travel history differs from the first record(colored in red). This can be said that this card was duplicated in m87.

Above are the types of fraud that can be detected.

Condition3-2/Condition3-3:

If transit card is duplicated after the previous record’s departure, and somehow was managed to get back to the current station without leaving a travel record(using another TransitCard for departure, while carrying the duplicated card as well) this cannot be detected.

```
{"station":"betelgeuse","passenger":{"id":"bda162ae-f1d3-48d0-
a031-87cec14cf2e9","florples":97,"history":
[{"date":"2019-06-15T20:36:50.460892Z","change":-3,"station":"betelgeuse"},
{"date":"2019-06-11T22:51:24.460892Z","change":100,"station":"betelgeuse"}],
"hash":"2dCm38rcoQSs0cn+h+/OdARn0tMdhcDeCfki1GLwF3k\u003d"}}

{"station":"betelgeuse","passenger":{"id":"bda162ae-f1d3-48d0-
a031-87cec14cf2e9","florples":97,"history":
[{"date":"2019-06-16T20:36:50.460892Z","change":-3,"station":"betelgeuse"},
{"date":"2019-06-15T21:36:50.460892Z","change":3,"station":"betelgeuse"},
{"date":"2019-06-15T20:36:50.460892Z","change":-3,"station":"betelgeuse"},
{"date":"2019-06-11T22:51:24.460892Z","change":100,"station":"betelgeuse"}],
"hash":"pt6R3eejk3Crq+DO/eEqbTZuqJzDfDR0lvOpa3nIbSo\u003d"}}
```

Above, we can see that this person departed from Betelgeuse on June-15 and visited Betelgeuse again to depart on June-16. For this type of record, we do not know whether this person’s card was duplicated after the departure of Betelgeuse on June-15, and then somehow brought the card back to Betelgeuse without leaving travel history, carrier of the card could have simply used another transit card to get to Betelgeuse OR just went for a sightseeing tour on June-15 and returned back to Betelgeuse.

Hence, I did not flag an alert on these kinds of cases.

It would have been better to scan only the inter-stationary travels, in order to examine each record more meticulously, catching more fraudsters.

2. One paragraph describing the approach

I tried to catch the fraudulent cards using the above conditions, which comes down to if-else statements. In order to compare with the previous record of the same ID, I utilized HashMap<UUID,TransitCard> to save the most recent *legit* transactions for given id as a key. Also, if it is found to be fraudulent, I have saved those passenger IDs in a blacklist in the form of an element in a HashSet. In order to block all passenger ID with the fraudulent record, I checked if the passenger ID is in the blacklist for every incoming departureRecord, if it does, it is flagged as a violation. Also, when comparing two different transaction histories, I have hashed both TranitCard Object and saw if hashes have equal value instead of iterating on each record.

3.Validation of Source Code.

I have made TestSpaceStationSecurity.java to test the examples stated in the section 1 and added condition1.gz, condition2.gz, condition3-1.gz, condtiion3-2.gz, condition3-3.gz in the data folder.