**System Test case Document for Backend**

Tested again with new document

Date: 2.6.2016

Performed by (Chrome):

Mzeen @ Acer Laptop, Windows 10

Started SpringBootApp in 13.745 seconds (JVM running for 14.662)

Tested on PRAKTIKUM01 again

Date: 3.6.2016

Performed by ricarda42

| **Use Case** | **Function Being Tested** | **Initial System State** | **Input** | **Expected Output** | **Test execution documentation** |
| --- | --- | --- | --- | --- | --- |
| IC\_C1 | SF\_S3: Send list of stops | Stop table exists in database | ./stops | Send the list of stops with stopID, stop name, lines passing through the stops, location of the stop, bus schedule of the stop and timestamp | List of stops with stopID, stop name, line passing through the stops, location of the stop, bus schedule of the stop is displaying. Schedule contains line name, lineID, stopID, arriving time and timestamp. |
| IC\_C2 | SF\_S5: Send properties of bus |  |  | *Not implemented in this iteration. SF\_S2 can be used instead (see below)* |  |
|  | SF\_S6: Send GPS-Data | GPS data was sent to database | ./busses/{busId}/  *(In interface doc: ./busses/:{busId})* | Send Current GPS Data coordinates with timestamp | Bus GPS coordinates shown with timestamp |
| IC\_C3 | SF\_S6: Send GPS-Data |  |  | See above |  |
| IC\_B1 | SF\_S1: Send list of lines | Line table exists in database | ./lines | Send List of Lines with properties lineID, name, routeID, timestamp and busses running on that specific line | List of Lines with lineID, name, routeID, timestamp and busses running on that specific line is displayed |
|  | SF\_S2: Send list of busses | Bus table exists in database | ./busses | Send List of all busses with their busID, number plate, color, picture and the line they serve | List of busses containing busID, number plate, the line served, color and picture is displayed |
|  | SF\_S4: Send list of routes | Route table exists in database | ./routes | Send List of all routes with routeID, route coordinates in geojson format and timestamp | List of all routes with routeID, route coordinates in geojson format and timestamp |
|  | SF\_S7: Store bus and line | Bus and line tables exist in database | ./updateBusStatus  Method: POST  Content type: application/json  Content:  {  "lineId":number,  "busId":number  } | Bus and line information stored | Tested with lineID = 3, busID = 2, with initially bus 2 on line 1. Output of post is 200 OK, calling ./lines reveals that bus 2 is now on line 3. |
| IC\_B2 | SF\_S13: Store number of available seats |  |  | *For next iteration* |  |
| IC\_B3 | SF\_S4: Send list of routes |  |  | See above |  |
| IC\_B4 | SF\_S3: Send list of stops |  |  | See above |  |
| - | SF\_S8: Store GPS data | Bus table exists in database | ./realTimeData  Method: POST  Content type: application/json  Content:  {  "busId":number,  “position”: { “type”: “Point”, “coordinates”: [double, double]},  “timestamp”: timeStamp  } | GPS Data of bus is stored | With busID = 1, coordinates [15.0, -3.4] and timeStamp = 1098692636000  the output of POST is 200 Ok, calling ./busses/1 reveals that coordinates have been changed to [15.0, -3.4]. The timeStamp remains 1098692634000. |
| - | SF\_S9: Send latest timestamp | Line, bus, route and stops tables exist in database | ./update | Last timestamps for bus, route, line and stops | Displays timestamp for bus, route, line and stops |

**Author:** Muhammad Zeeshan

**Status:** complete

**Reviewer:** Ricarda Rosemann

**Review Status:** incomplete