Test cases Document for Backend testing

System test cases of “Test cases.pdf” reviewed and tested again on PRAKTIKUM01.
Date: 3.6.2016
Last modifications: 7.6.2016
Performed by ricarda42
Version: 0.1 in development, on commit 5c6fd65

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	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.	Passed
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}/	Current GPS Data coordinates with timestamp	Passed
IC_C3	SF_S6: Send GPS-Data			See above	
IC_B1	SF_S1: Send list of lines		./lines	List of Lines with properties lineID, name, routeID, timestamp and busses running on that specific line	Passed
	SF_S2: Send list of busses	Bus table exists in database	./busses	List of all busses with their busID, number plate, color, picture and the line they serve	Passed
	SF_S4: Send list of routes	Route table exists in database	./routes	List of all routes with routeID, route coordinates in geojson format and timestamp	Passed
	SF_S7: Store bus and line	Bus and line tables exist in database	./updateBusStatus Method: POST	Output of post 200, Bus and line information stored	Passed; tested with lineID = 3, busID = 2, with initially bus 2 on line 1. Output of

			Content type: application/json
			Content:
			{
			"lineId":number
			,
			"busId":number
			}
					post is 200, 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
			}
				Output of post 200, GPS Data of bus is stored, timestamp not changed	Passed;
					With busID = 1, coordinates [15.0, -3.4] and timeStamp = 1098692636000
					the output of POST is 200, 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	Passed

Author: Muhammad Zeeshan
Status: complete
Reviewer: Ricarda Rosemann
Review Status: Complete