Test cases Document for Backend testing – Iteration 4

Extended System test cases of “Test Cases_reviewed.pdf”
Date: 7.6.2016
Last modifications: 16.7.2016
Performed by Mohammad Baniasad
Version: 0.2 in development, on commit 449e3b5

Tests marked with *Redone:* Redone after implementation of temporary work-around to comply with interface documentation
Performed by ricarda42
Date 26.6.2016
Version: commit b07e6de in branch developmentIt3

System test cases extended for iteration 4; redone tests from iteration 3 also to be tested with cleaned-up code

added automated tests with SOAPUI, the initial system data would be as expected entered in /home/moo/Documents/uni/project/SS2016-group1/buergerbusserver/core/target/classes/db/migration/V0002__R001_Master_data.sql

Sr. no	Use Case	Function Being Tested	Initial System State	Input	Expected Output	Test execution documentation	Automated
1	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	Getting saved data - stops
2	IC_C2	**SF_S5: Send properties of bus**			*Not needed*		
3		**SF_S6: Send GPS-Data**	GPS data was sent to database	./busses/{busId}/	Current GPS Data coordinates with timestamp	Passed	Getting saved data - Busses- single bus
4	IC_C	**SF_S6: Send**			See above		

	3	**GPS-Data**					
5	IC_B 1	**SF_S1: Send list of lines**		./lines	List of Lines with properties lineId, name, routeId, timestamp and busses running on that specific line	Passed	Getting saved data - Lines
6		**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	Getting saved data - Busses -all busses
7		**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	Getting saved data - Routes -all routes

8		**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
				}
					Result status 200,
					Bus and line information stored
						Passed	Setting bus – update bus status

9	IC_B 2	**SF_S13: Store number of available seats (see No. 12 – identical interface)**		./realTimeData
				
				Method:POST
				Content type: application/json
				Content:
				{
				“busId”:number,
				“position”: { “type”: “Point”,
				“coordinates”: [double, double]},
				“takenSeats”: number
				“timestamp”: timeStamp
				}
					Result status 200;
					GPS data and taken seats stored
						passed	Setting bus - update bus position and seats

10	IC_B	**SF_S4: Send list**			See above		

	3	**of routes**					
11	IC_B 4	**SF_S3: Send list of stops**			See above		

12	-	**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
				}
					Result status 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.	Setting bus - update bus position and seats

13	-	**SF_S9: Send latest timestamp**	Line, bus, route and stops tables exist in database	./update	Last timestamps for bus, route, line and stops	Passed	Getting saved data -Last updates
14	IC_B 5	**SF_S10: Notification of**	CitizenApp sends new	./customStops	Result status 200;	Passed	Custom stops – create and load to

		custom stop request
		
		Post custom stop request
			Custom Stop Request	Method: POST
				Content type: application/json
				Content:
				{
				"lineId":number,
				"location": { "type": "Point",
				"coordinates": [double , double] },
				"numberOfPersons": number,
				"deviceId":String,
				"info": {
				"name": "string",
				"address": "string",
				"assistance": [number,number...]
				}
				"pickUpTime":Timestamp
				}
					New custom stop with assigned requestId, lineId, pickUpTime, location, number of persons, deviceId, information of passenger included name, address, and assistance, status and timestamp		check

15.1		**Get custom stop requests ref. to line** **Valid 1**	Request from CitizenApp with lineId stored where global status is 1. DriverApp requests customStops for given lineId and busId. The bus has not yet rejected the custom stop (i.e. busId not in rejectingBus	./customStops?lineId={lineId}&busId={busId}	Custom stop request included in output list		

			list)				
15.2		**Get custom stop requests ref. to line Invalid 1**	Request from CitizenApp with lineId stored where global status is 1. DriverApp requests customStops for given lineId and busId. The bus has already rejected the custom stop (i.e. busId in rejectingBus list)	./customStops?lineId={lineId}&busId={busId}	Custom stop request not included in output list		
15.3		**Get custom stop requests ref. to line Valid 2**	Request from CitizenApp with lineId stored. DriverApp requests customStops for given lineId and busId. Global status is 2. The bus has accepted this custom stop (i.e. acceptingBus is busId).	./customStops?lineId={lineId}&busId={busId}	Custom stop request included in output list.		
15.4		**Get custom stop requests ref. to line Invalid 2**	Request from CitizenApp with lineId stored. DriverApp requests customStops for	./customStops?lineId={lineId}&busId={busId}	Custom stop request not included in output list.		

			given lineId and busId. Global status is 2. The bus has accepted this custom stop (busId is not acceptingBus)				
15.5		**Get custom stop requests ref. to line Invalid 3**	Request from CitizenApp with lineId stored. DriverApp requests customStops for given lineId and busId. All busses on this line rejected this custom stop, i.e. global status is 3.	./customStops?lineId={lineId}&busId={busId}	Custom stop not in output list		
15.6		**Get custom stop requests ref. to line Invalid 4**	Request from CitizenApp with lineId stored. DriverApp requests customStops for given lineId and busId. A bus accepted and completed this custom stop, i.e. global status is 4.	./customStops?lineId={lineId}&busId={busId}	Custom stop not in output list		
15.7		**Get custom stop requests ref. to line Invalid 5**	Request from CitizenApp with lineId stored. DriverApp	./customStops?lineId={lineId}&busId={busId}	Custom stop not in output list		

			requests customStops for given lineId and busId. A bus accepted this custom stop, but the citizen did not show up, i.e. global status is 5.				
15.8		**Get custom stop requests ref. to line Valid 3**	Request from CitizenApp with lineId stored. DriverApp requests customStops for given lineId and busId. This bus accepted this custom stop, but the citizen canceled it, i.e. global status is 6 and busId is acceptingBus.	./customStops?lineId={lineId}&busId={busId}	Custom stop is in output list		
15.9		**Get custom stop requests ref. to line Invalid 6**	Request from CitizenApp with lineId stored. DriverApp requests customStops for given lineId and busId. A bus different from this bus accepted this custom stop, but the citizen	./customStops?lineId={lineId}&busId={busId}	Custom stop is not in output list		

			canceled it, i.e. global status is 6 and busId is not acceptingBus.				
15. 10		**Get custom stop requests ref. to line Invalid 7**	DriverApp requests pending customStops for given lineId and busId; no requests for this line	./customStops? lineId={lineId}&busId={busId}	Empty list		
15. 11		**Get custom stop requests ref. to line Invalid 6**	DriverApp requests pending customStops for given lineId and busId; requests for that line have pickUpTime before current time	./customStops? lineId={lineId}&busId={busId}	Empty list		

16. 1		**SF_S11: Response to custom stop request**
		
		Update status of custom stop requests Valid 1
			DriverApp of a given bus received custom stop requests for the given line.
			The driver accepts a custom stop with status acceptedb.
				./customStops/ {requestId}
				
				Method: POST
				Content type: application/json
				Content:
				{
				"status":2
				“busId”: number
				}
					Result status 200; Status of given request set to 2, acceptingBus set to busId	BusId does not exist	Accepting bus works.
							
							Custom stops-create check accept check

16. 2		**Update status of custom stop requests**	DriverApp of a given bus received custom	./customStops/ {requestId}	Result status 200; status of request unchanged		

		Invalid 1	stop requests for the given line.
			The driver accepts a custom stop where the status has already been changed.
				Method: POST
				Content type: application/json
				Content:
				{
				"status":2
				“busId”: number
				}			

16.3		**Update status of custom stop requests**
		Valid 2
			DriverApp of a given bus received custom stop requests for the given line.
			The driver rejects a custom stop with status pending. Not all busses on this line have already rejected this custom stop.
				./customStops/{requestId}
				
				Method: POST
				Content type: application/json
				Content:
				{
				"status":3
				“busId”: number
				}
					Result status 200; status of request unchanged, busId added to list rejectingBus		

16.4		**Update status of custom stop requests**
		Valid 3
			DriverApp of a given bus received custom stop requests for the given line.
			The driver rejects a custom stop with status pending. All busses on this line have already rejected
				./customStops/{requestId}
				
				Method: POST
				Content type: application/json
				Content:
				{
				"status":3
				“busId”: number
				}
					Result status 200; status of request set to 3, busId added to list rejectingBus		

			this custom stop.				

16.5		**Update status of custom stop requests Valid 4**	DriverApp of a given bus received custom stop requests for the given line. The driver rejects a custom stop where the status has already been changed.	./customStops/{requestId}
				
				Method: POST
				Content type: application/json
				Content:
				{
				"status":3
				“busId”: number
				}
					Result status 200; status of request unchanged, busId added to rejectingBus list		

16.6		**Update status of custom stop requests Valid 5**	DriverApp of a given bus received custom stop requests for the given line. The driver accepted a custom stop (i.e. status is 2 and acceptingBus equals busId) and now marks it as complete.	./customStops/{requestId}
				
				Method: POST
				Content type: application/json
				Content:
				{
				"status":4
				“busId”: number
				}
					Result status 200; status of request set to 4		

16.7		**Update status of custom stop requests Invalid 2**	DriverApp of a given bus received custom stop requests for the given line. A driver different from this driver accepted a	./customStops/{requestId} Method: POST Content type: application/json Content: { "status":4	Result status 200; status of request unchanged		

			custom stop (i.e. status is 2 and acceptingBus is not equal to busId) This bus driver now marks this custom stop as complete.	“busId”: number }			

16.8		**Update status of custom stop requests Invalid 3**	DriverApp of a given bus received custom stop requests for the given line.
			This driver accepted a custom stop but the status was updated further (i.e. status is not 2 and acceptingBus is equal to busId).
			This bus driver now marks this custom stop as complete.
				./customStops/{requestId}
				
				Method: POST
				Content type: application/json
				Content:
				{
				"status":4
				“busId”: number
				}
					Result status 200; status of request unchanged		

16.9		**Update status of custom stop requests Invalid 4**	DriverApp of a given bus received custom stop requests for the given line. The bus driver now marks a custom stop as	./customStops/{requestId} Method: POST Content type: application/json Content: { "status":4	Result status 200; status of request unchanged		

			complete he did not accept and where the status is not accepted.	“busId”: number }			

16.10		**Update status of custom stop requests**
		Valid 6
			DriverApp received custom stop requests for the given line.
			The driver accepted a custom stop (i.e. status is 2 and acceptingBus equals busId) and now marks it as not shown up.
				./customStops/{requestId}
				
				Method: POST
				Content type: application/json
				Content:
				{
				"status":5
				“busId”: number
				}
					Result status 200; status of request set to 5		

16.11		**Update status of custom stop requests**
		Valid 7
			DriverApp of a given bus received custom stop requests for the given line.
			The driver accepted a custom stop (i.e. status is 2 and acceptingBus equals busId) and now marks it as not shown up.
				./customStops/{requestId}
				
				Method: POST
				Content type: application/json
				Content:
				{
				"status":5
				“busId”: number
				}
					Result status 200; status of request set to 5		

16.12		**Update status of custom stop**	DriverApp of a given bus	./customStops/{requestId}	Result status 200; status of request unchanged		

		requests Invalid 5	received custom stop requests for the given line. A driver different from this driver accepted a custom stop (i.e. status is 2 and acceptingBus is not equal to busId) This bus driver now marks this custom stop as not shown up.	Method: POST Content type: application/json Content: { "status":5 “busId”: number }			
16. 13		**Update status of custom stop requests Invalid 6**	DriverApp of a given bus received custom stop requests for the given line. This driver accepted a custom stop but the status was updated further (i.e. status is not 2 and acceptingBus is equal to busId). This bus driver now marks this custom stop as not shown up.	./customStops/ {requestId} Method: POST Content type: application/json Content: { "status":5 “busId”: number }	Result status 200; status of request unchanged		
16. 14		**Update status of custom stop**	DriverApp of a given bus	./customStops/ {requestId}	Result status 200; status of request unchanged		

		requests Invalid 7	received custom stop requests for the given line. The bus driver now marks a custom stop as not shown up he did not accept and where the status is not accepted.	Method: POST Content type: application/json Content: { "status":5 “busId”: number }			
16. 15		**Update status of custom stop requests Valid 8**	DriverApp of a given bus received custom stop requests for the given line. The citizen cancels the custom stop.	./customStops/ {requestId} Method: POST Content type: application/json Content: { "status":6 “busId”: number }	Result status 200; status of request set to 6		
16. 17		**Update status of custom stop requests Valid 9**	DriverApp of a given bus received custom stop requests for the given line. The citizen cancels the custom stop.	./customStops/ {requestId} Method: POST Content type: application/json Content: { "status":6 }	Result status 200; status of request set to 6		
18. 1		**Get custom stop requests ref. to requestId and deviceId**	CitizenApp has the requestId	./customStops? requestId={request Id}&deviceId={devi ceId}	Custom stop request for the given requestId with all properties	*Redone:* Passed	

18.2		**Get custom stop requests ref. to requestId**	CitizenApp has the requestId	./customStops?requestId={requestId}	Custom stop request for the given requestId with all properties		
18.3		**Get custom stop requests ref. to deviceId**	CitizenApp has the requestId	./customStops?deviceId={deviceId}	All custom stops for the given deviceId with all properties		
19	IC_B6	**SF_S11: Response to custom stop request**	DriverApp send status as complete	*See above with status complete*			

Author: Ricarda Rosemann
Status: Complete
Reviewer:
Review Status: