**Test cases Document for Backend testing**

Extended System test cases of “Test Cases\_reviewed.pdf”

Date: 7.6.2016

Last modifications: 22.6.2016

Performed by mzeen

Version: 0.2 in development, on commit 449e3b5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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  Content type: application/json  Content:  {  "lineId":number,  "busId":number  } | Result status 200,  Bus and line information stored | Passed;  tested with lineID = 3, busID = 2, with initially bus 2 on line 1. Output of 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  } | 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. |
| - | SF\_S9: Send latest timestamp | Line, bus, route and stops tables exist in database | ./update | Last timestamps for bus, route, line and stops | Passed |
| IC\_B5 | SF\_S10: Notification of custom stop request | CitizenApp sends new Custom Stop Request | ./customStops/newRequest  Method: POST  Content type: application/json  Content:  {  "lineId":number,  "lat":double,  "lon":double,  "numberOfPersons":number,  "deviceId":String,  "userName":String,  "userAddress":String,  "userAssistance":[number,number, …],  "pickUpTime":Timestamp  } | Result status 200;  New custom stop with assigned requestId, lineID, pickUpTime, location, number of personson, deviceID, information of passenger included name, address, and assistance, status and timestamp |  |
|  |  | Request from CitizenApp with lineId has been stored, DriverApp requests pending customStops for given lineId | ./customStops?lineId={lineId} | All custom stop requests for the given line with pickUpTime after current time and status pending, each with all their properties (see above) |  |
|  |  | DriverApp requests pending customStops for given lineId; no requests for this line | ./customStops?lineId={lineId} | Empty list |  |
|  |  | DriverApp requests pending customStops for given lineId; requests for that line have pickUpTime before current time | ./customStops?lineId={lineId} | Empty list |  |
|  | SF\_S11: Response to custom stop request | DriverApp received custom stop requests for the given line, sends an update of status on a custom stop request | ./customStop/updStatus  Method: POST  Content type: application/json  Content:  {  "requestId":number,  "status":number  } | Result status 200; Status of given request updated as given |  |
|  |  | CitizenApp has the requestId | ./customStop?requestId={requestId}&deviceId={deviceId} | Custom stop request for the given requestId with all properties |  |
| IC\_B6 | SF\_S11: Response to custom stop request | DriverApp send status as complete | *See above with status complete* |  |  |

**Author:** Muhammad Zeeshan, Ricarda Rosemann

**Status: C**omplete

**Reviewer:**

**Review Status:**