***Public Transport System Management***

**IS 301 Enterprise Integration**

***G2-3***

**Assignment**

***EUGENE CHOY WEN JIA***

***HO MIN KIT WINSTON***

***HO WEI HONG***

***SIM LI JIN***

***YIN YUKUN***

***YONG FU XIANG***

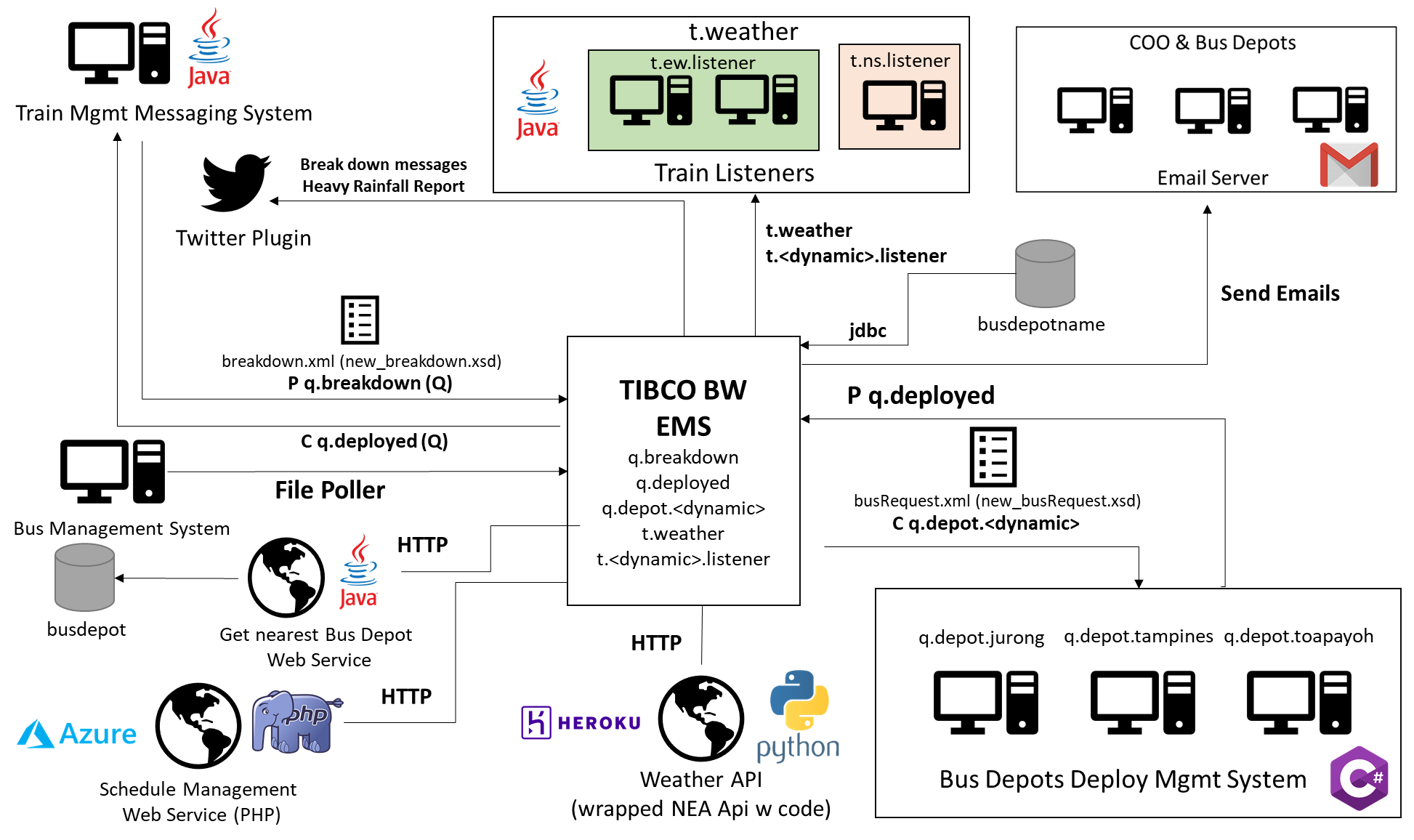
# Introduction

The Public Transport network in Singapore serves hundreds of thousand passengers every day. Any disruption in the service could be disastrous and mitigation actions must be undertaken to ensure minimal disruptions to the passengers. Integrations of various system in the Public Transport Network is therefore important to help enhance communication and automate processes if mitigation actions are required.

On top of having to mitigate disruptions, there are also other processes such as checking the weather to determine the speed of travel and the bus schedules of bus drives which requires several steps before the message can be transferred from one end to the other. Using integration tools, we can automate a huge bulk of this process reducing the time needed for human intervention.

# Business Scenario

## Technical Overview Diagram

**

## Train Breakdown Process

## Weather Reporting Process

## Bus Schedule Process

# JMS Interactions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| JMS Integration | From | To | \*FF/  SRR/  ARR | Publish-Subscribe or Point-to-Point | JMS Queue/Topic (or Topic with durable subscription) | Queue/Topic Name used (case-sensitive) |
| Integration 1 | Train Management System | Tibco IM | SRR | Point-to-Point | JMS Queue | q.breakdown |
| Tibco IM | Train Management System | Point-to-Point | JMS Queue | q.deployed |
| Integration 2 | Tibco IM | Bus Depot <Dynamic> | FF | Point-to-Point | JMS Queue | q.depot.<dynamic> |
| Integration 3 | Train Management System | Tibco IM | FF | Point-to-Point | JMS Queue | q.resumed |
| Tibco IM | Bus Depot <Dynamic> | Point-to-Point | JMS Queue | q.depot.<dynamic> |
| Integration 4 | Tibco IM | Train Listener | FF | Publish-Subscribe | Topic | t.weather |

**Legend** FF - Fire & Forget SRR - Sync Request/Reply ARR - Async Request/Reply

# Web Services

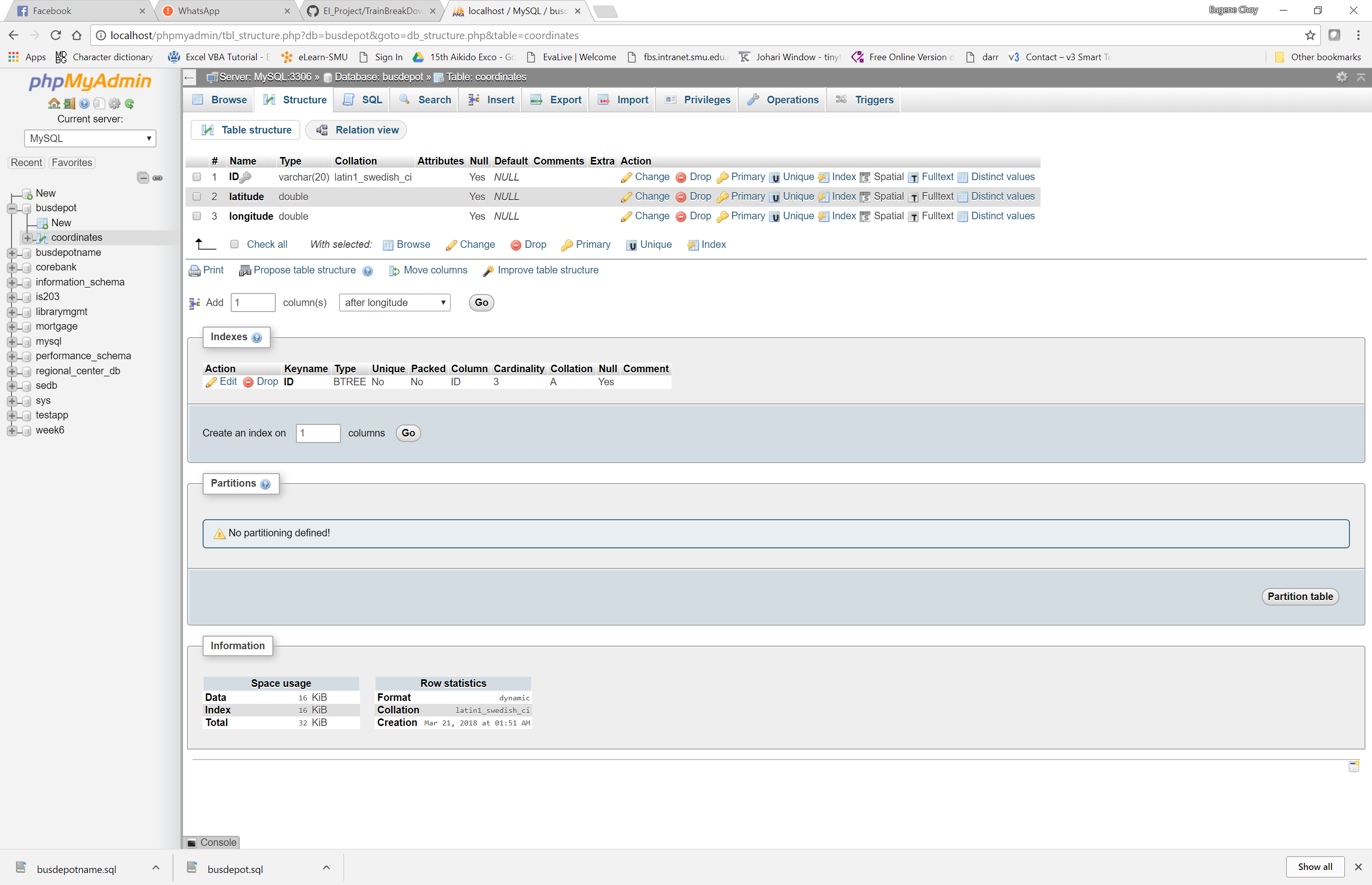
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service | Description | Protocol | Input | Output |
| *Get Nearest Bus Depot Web Service (Java)* | *A transformed XML created from the breakdown.xml is sent into the web service as XML String to retrieve the depot ID*  *Self-Coded in Java* | REST POST | *XML*  *Train id*  *Fault Type*  *Time stamp*  *Coordinates* | *Depot ID* |
| *Schedule Polling* | *Upload a txt file onto the Schedule Poller service which can be downloaded by other users*  *Self-Coded in PHP* | REST POST | *Text File*  *Multi Part* | *JSON*  *URL of the file*  *Status Code* |
| *Weather API* | *Check the rainfall in Singapore from NEA web service*  *NEA API + Self-Coded wrapper in Pythong (Self-coded wrapper randomize the chance of getting a rain fall if it is not having a rainfall in Singapore now – For Demo purpose)* | REST GET | *-* | *XML*  Heavy Rain Warning |

# Design/Schema and Content of Data

## Database

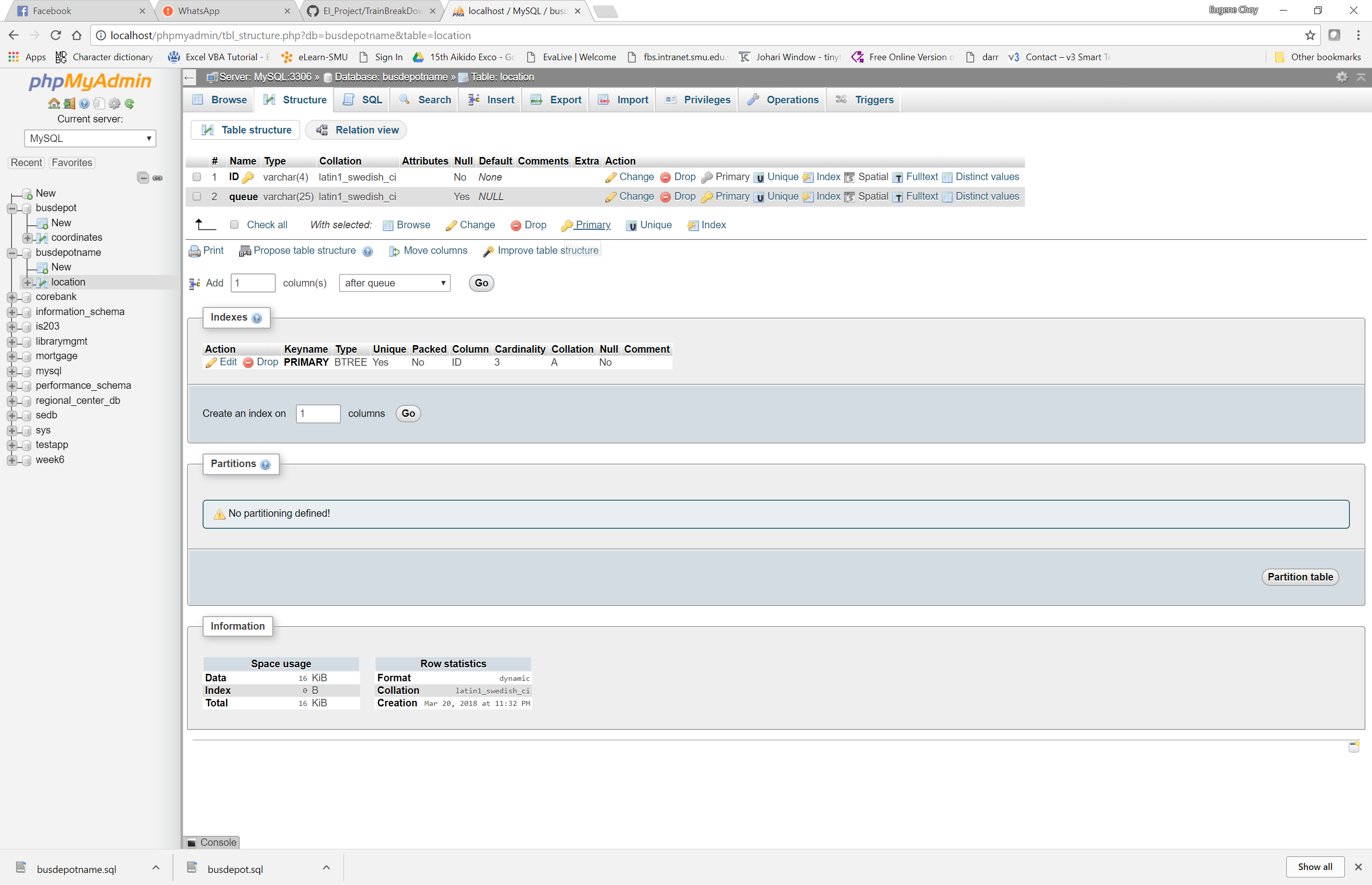
### Bus Depot

**coordinate table**



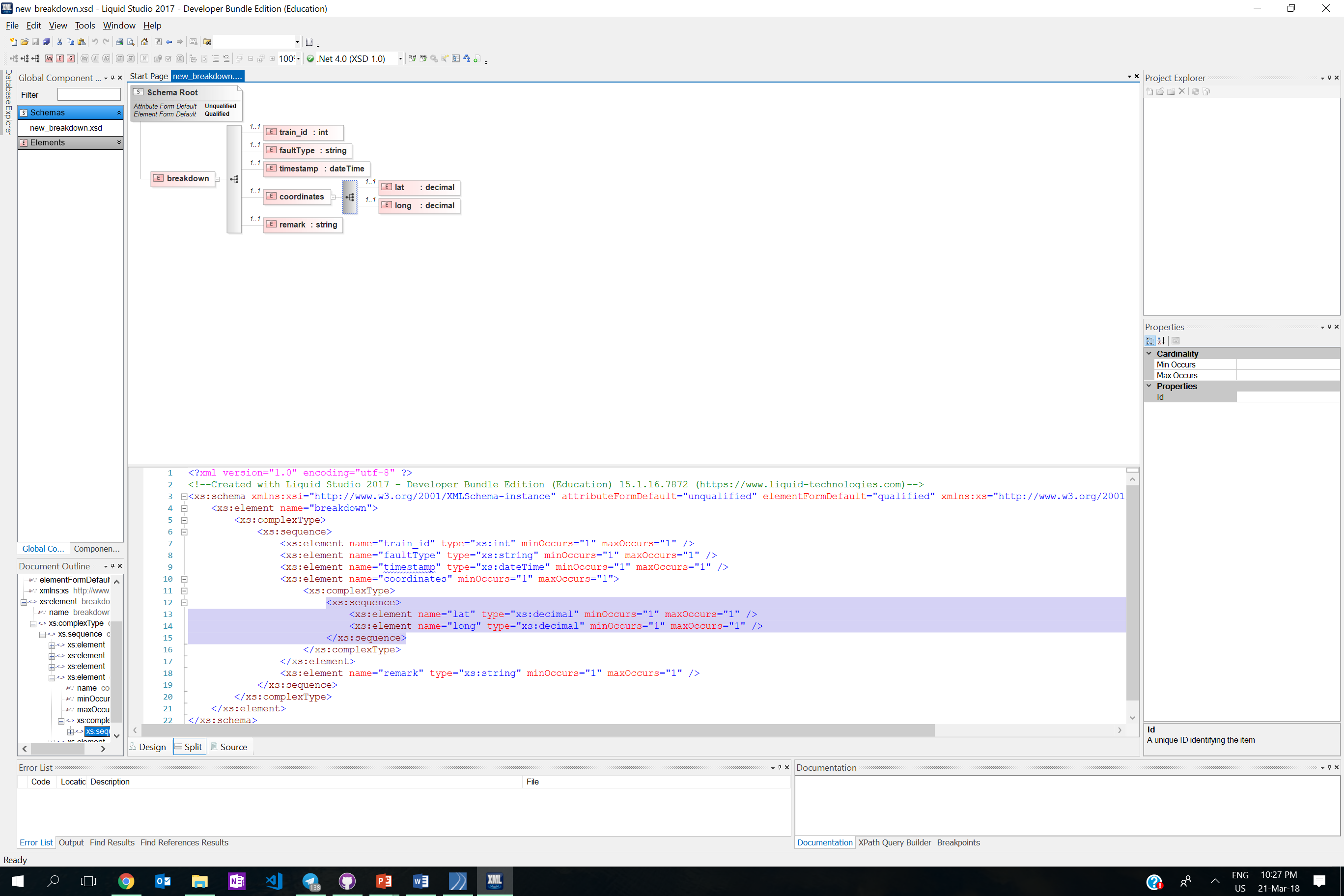
### Bus Depot Name

**location table**



## XSD

### new\_breakdown.xsd



*Sample:*

<?xml version = **"1.0"** encoding = **"UTF-8"**?>

<breakdown xmlns:xsi = **"http://www.w3.org/2001/XMLSchema-instance"** xsi:noNamespaceSchemaLocation = **"..\\new\_breakdown.xsd"**>

<train\_id>**3**</train\_id>

<line>**ew**</line>

<faultType>**Train Fault**</faultType>

<timestamp>**2018-02-20T09:00:00**</timestamp>

<coordinates>

<lat>**1.3272383**</lat>

<long>**103.9443528**</long>

</coordinates>

<remark>**Tanah Merah Station**</remark>

</breakdown>

### new\_breakdown\_formatted.xsd



Sample:

<?xml version=**"1.0"** encoding=**"UTF-8"**?>

<breakdown>

<train\_id>**3**</train\_id>

<faultType>**Train Fault**</faultType>

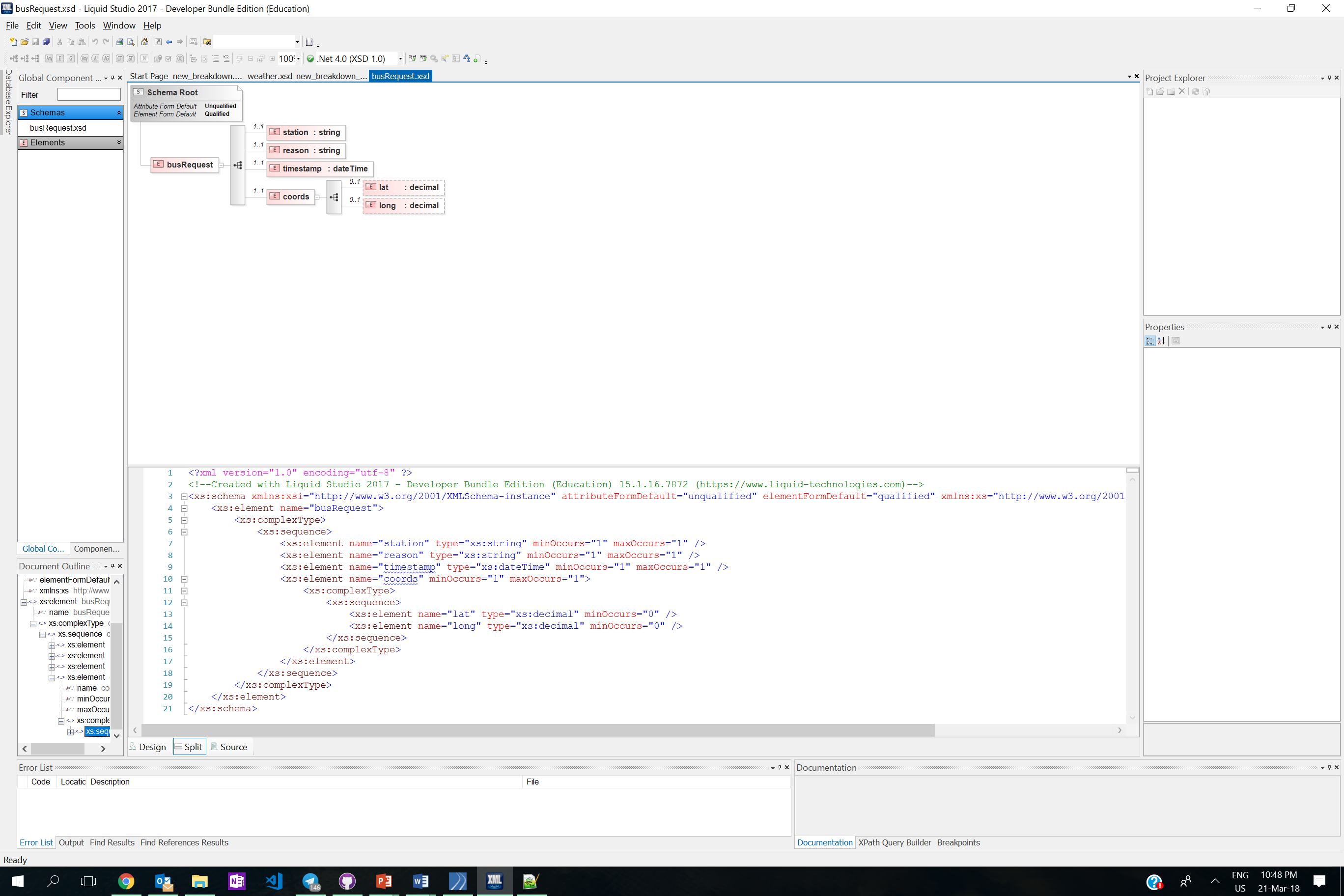
<timestamp>**2018-02-20T09:00:00**</timestamp>

<coordinates>**1.3272383,103.9443528**</coordinates>

<remark>**Tanah Merah Station**</remark>

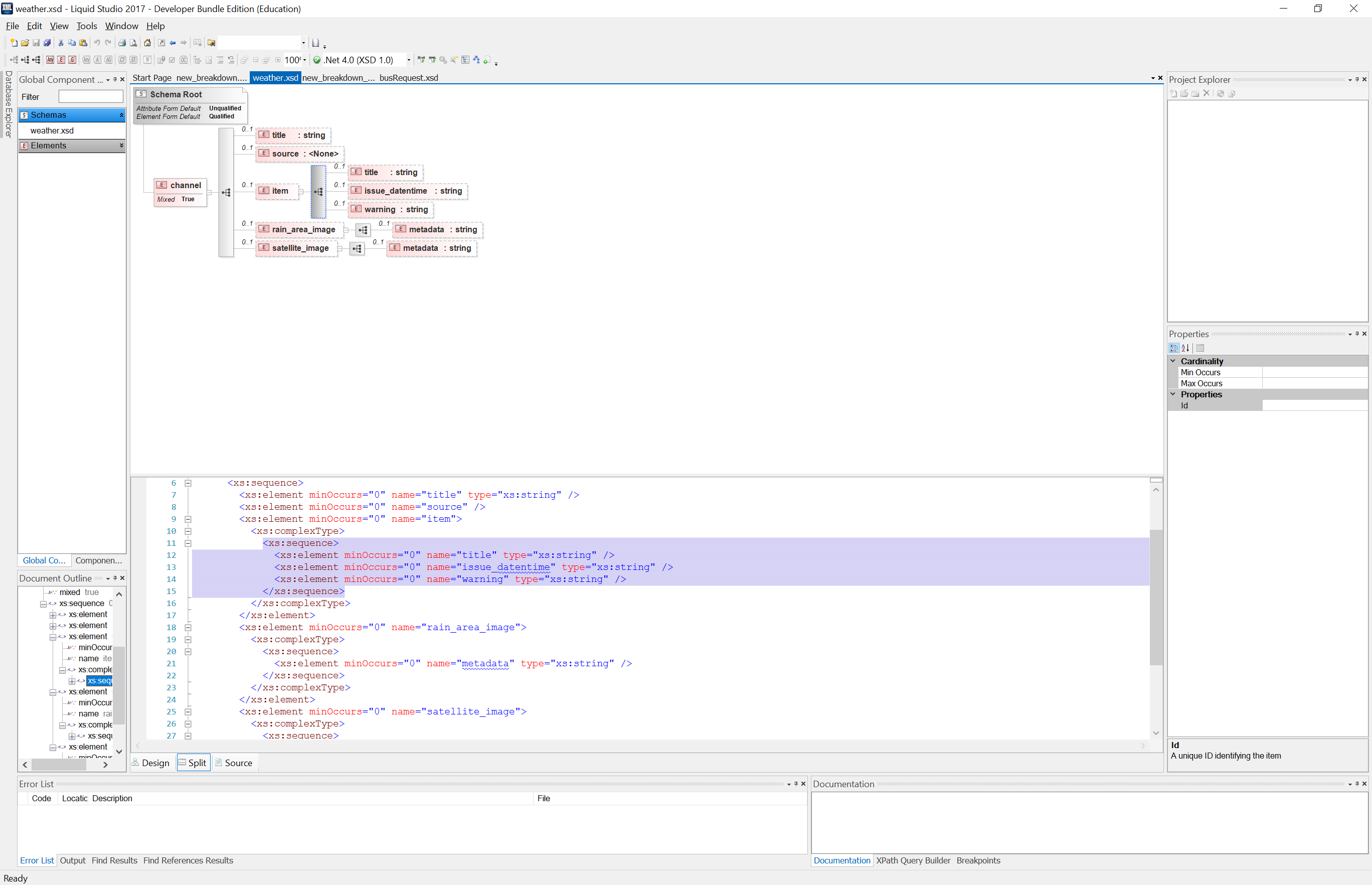
</breakdown>

### busRequest.xsd



Sample:

### weather.xsd



Sample

<channel>

<title>**Heavy Rain Warning**</title>

<source>**Meteorological Service Singapore** </source>

<item>

<title>**HEAVY RAIN WARNING**</title>

<issue\_datentime>**-**</issue\_datentime>

<warning>**The heavy rain has eased. Thundery showers clearing in the evening.**</warning>

</item>

<rain\_area\_image><metadata>**null**</metadata></rain\_area\_image>

<satellite\_image><metadata>**null**</metadata></satellite\_image>

</channel>

## JSON

### JSON from Schedule Polling

This JSON is returned upon successful upload of a file to the server

{

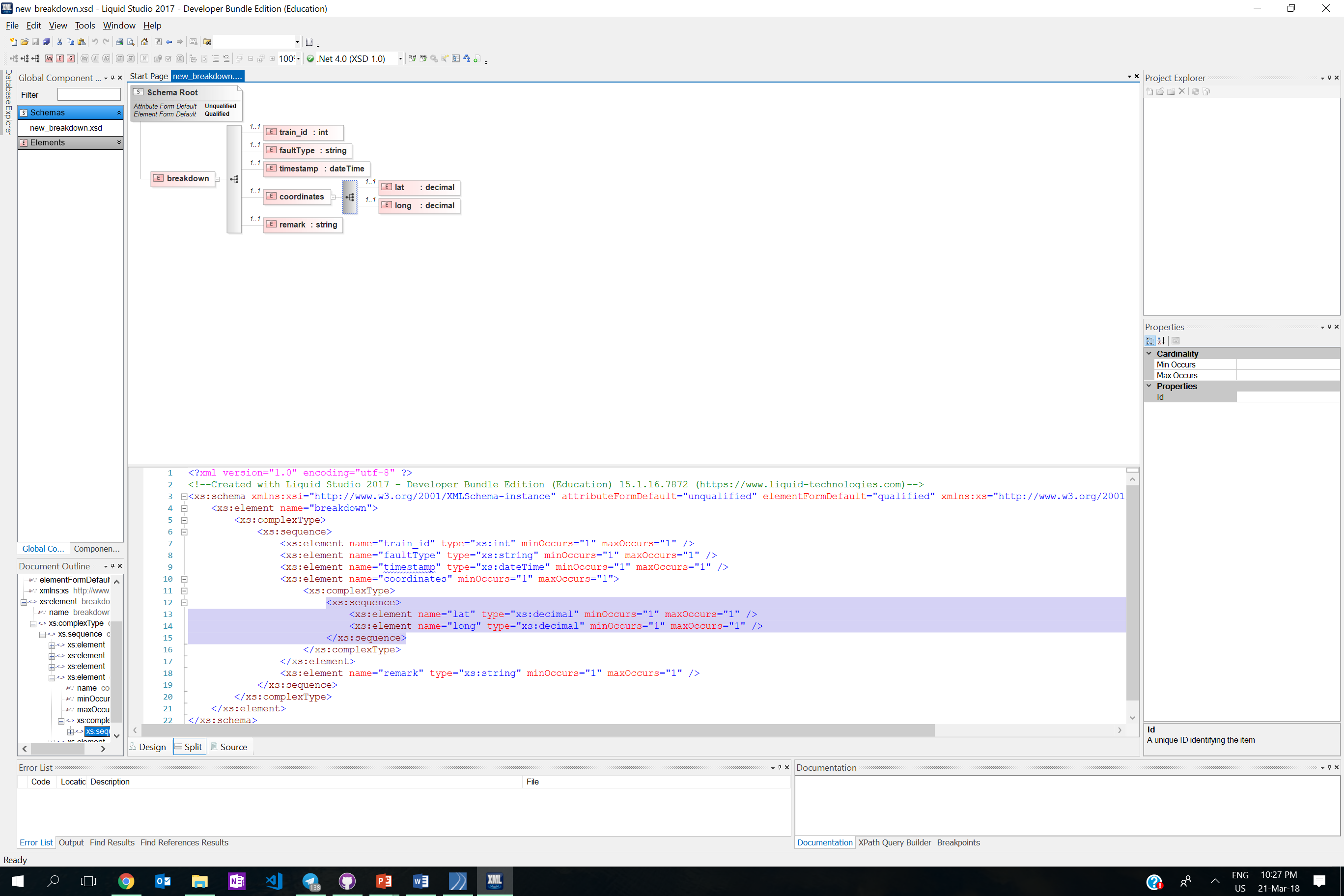
"status":"ok",   
"statusCode":"200",   
"link":”https:\/\/eieio.blob.core.windows.net\/schedule\/January2018Schedule.txt"   
}

### JSON from getting nearest Bus Depot

{"message":"TMPN","status":"success"}

# Data Transformation

***Between Train Management System(new\_breakdown.xsd) and Get nearest Bus Depot web service (new\_breakdown\_formatted.xsd)***



***Between Train Management System (new\_breakdown.xsd) and Bus Depots (bus\_request.xsd)***

# Content-based Routing

*[For each content-based routing, indicate the condition(s) used, the method used for determining the route and the destination system(s)/service(s)]*

# Beyond the Labs

*[List, describe and explain the things you have done beyond the labs]*

# Scenario Walkthrough (Max 10 Pages)

*[Walkthrough of your demonstration, using screen captures. Screens captured must be viewable when the document is displayed at 100% zoom scale]*