**Demonstration Flow Design of T-CONNECT for IDTO Project**

* **Notes**:
  + Only demonstrate the trip from "BART" to “Tri\_Delta 300" with the connection at Pittsburg/Bay Point BART Station. Thus the MySQL tables of these two routes, including static and real-time, should be fully functional.
  + The transfer connection table can be skipped this time.
  + The main algorithm to trigger push notification is temporarily minimized to using simple time ahead or after.
  + Add a table “T-connect” to include information of the trips having requested connection protection service, which is defined by

**Table T-connect**

Table schema:

CREATE TABLE `T-connect` (

`user\_id`,

`trip\_create\_time`,

`trip\_origin`

`trip\_destination`

`event\_req\_time `,

`user\_token`,

`this\_route\_agency `,

`this\_route\_id `,

`this\_route\_arr\_stop `,

`next\_route\_agency `,

`next\_route\_id `,

`next\_route\_dep\_stop`,

`event\_status`,

PRIMARY KEY (`user\_id`, `event\_status`),

) ENGINE=InnoDB DEFAULT CHARSET=latin1

* + If cannot make it done in time, the above table can be temporarily substituted by a same format CSV file.
  + The code of `event\_status`:

|  |  |  |
| --- | --- | --- |
| Value  (‘zyxs’ decimal) |  | Note |
| 0000 | Request not starts |  |
| 0001 | Request start |  |
| 0011 | Approval by dispatcher |  |
| 0021 | Rejected by dispatcher | x is the digit stands for dispatch operation |
| 0031 | Time out at waiting list |
| 01x1 | The transfer bus has left after the requested holding. | y is the digit stands for bus reaction |
| 02x1 | The transfer bus has left without a holding service. |
| 1yx1 | Service Success according to passenger’s feedback | z is the digit stands for passenger’s feedback |
| 2yx1 | Service Failure according to passenger’s feedback |
| 3yx1 | Time out for passenger’s feedback |

* **Demo Flow for All Involved Parties**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Passenger | Server | Dispatch Center | Bus Driver |
| 10:01 | O/D Input, request trip planning |  |  |  |
| Confirm Trip to subscribe notification service | Receive and store (to MySQL) information of planned trip (\*) |  |  |
|  | Get ETA of BART at “Pittsburg/Bay Point”, 10:17 for example |  |  |
| 10:12 |  | <5 min. before arrival>  Update “event\_state” in the trip table |  |  |
|  |  | Receive new request and push a message dialog |  |
| 10:14 |  |  | Ask driver to hold until 10:20 by calling |  |
|  |  |  | Approve connection protection |
|  |  | Press “Approve” button |  |
|  | Update “event\_state” in the trip table; request a push notification |  |  |
| Receive the first push notification: “Upon your request, the Bus 301 will wait at XXX until 10:20am.” |  |  |  |
| 10:20 |  |  |  | Finish waiting and departing |
| 10:21 |  | <1 min. after departure>  Update “event\_state” in the trip table |  |  |
| Receive the second push notification: “The bus 301 on your trip has left from xxx.” |  |  |  |
| 10:22 | Answer the question on web “Was the last service successful?” |  |  |  |
|  | Update “event\_state” in the trip table |  |  |

* **Working Flow for System Components**

