JSON tree file name: citi-hackathon-9a169-default-rtdb-export.json

Context:

Due to time constraint, we chose to hardcode data into the flutter app. However, in the future, user and merchant data will be extracted from the database. This document will be a brief overview on the database design and how it will support the in-app functions. This database will be remote.

Preface:

All “issue” ids and “voucher” ids are to be randomized in the future. QR images are generated in-situ rather than stored locally on user’s phones/merchant’s systems.

Database:

citi-hackathon-9a169-default-rtdb

* Merchant (contains all merchant data)
  + Adidas
    - Merchant\_account (stores merchant profile info)
      * Contact\_no.
      * Display\_name
      * Location
      * Password (for log-in verification, query adidas where password == password)
      * Profile\_image
    - Merchant\_listings (stores information on vouchers created by merchant)
      * Voucher\_1357
        + Claimed: 0 (actively tracked and increased when users purchase voucher)
        + Expired: 11 (actively tracked and increased when users purchase voucher)
        + Issued: 11 (actively tracked and increased when users purchase voucher)
        + Limit: 50 (max limit to issues)
        + Listing\_status: “delisted” (tracks the listing status)
* Users
  + Bob
    - Transaction\_history
      * Issue\_1312
        + Price
        + Status: either bought or sold
    - User\_account
      * Age: 22 (for demographic tracking on merchant side)
      * Contact\_number
      * Display\_name
      * Password
      * Points: 292 (used as in-app currency to purchase vouchers, earned through voucher resale or converted from cash deposit)
      * User\_image
      * User\_savings: $197.3 (tracked based on voucher claims)
    - User\_listings (tracks user issue listings on the voucher marketplace)
    - User\_wallet (contains all the user voucher issues / voucher favourites and their status (favorited, issued, claimed, expired)
* Voucher\_issues (contains the vouchers issued and the merchant and user\_ids for validation purposes / acts as an intermediate table for many to many relationship management on NoSQL)
* Vouchers (table that contains information of vouchers listed, upon voucher delisting on merchant side, the respective voucher entry is deleted from table)