Documentation

PR-07: Laundry Management System

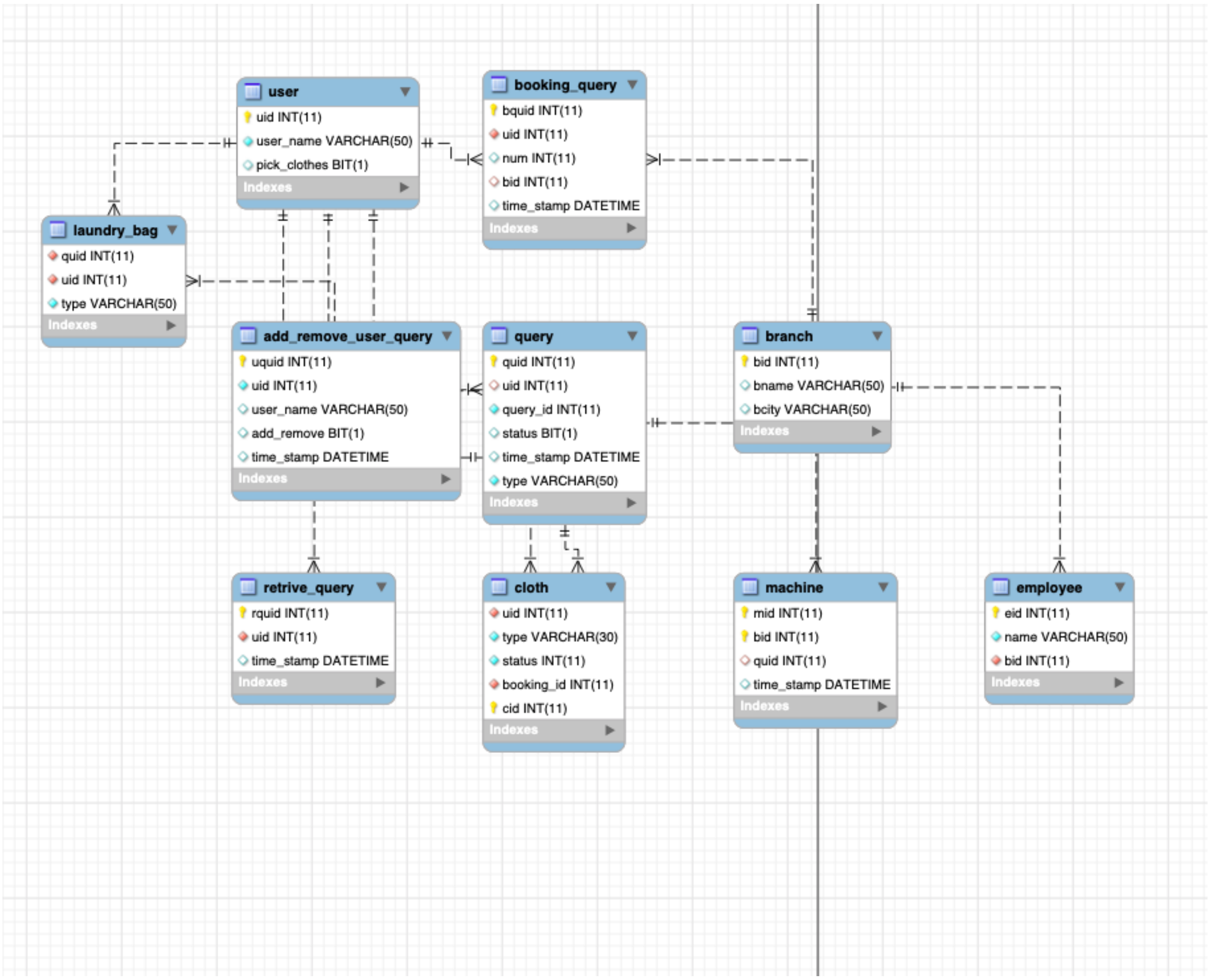
By
Ritik Upmanyu: 2019B3A70517P
Sahil Gupta: 2019B3A70154P

1. System requirement specification (SRS)

1. MySQL
2. Browser (to see frontend(already hosted))
3. If you need to compiling the frontend and backend code you'll need
    a. Node.js+express.js
    b. Git

2. System modeling

a. Entity-relationship (ER) diagram



b. Schema design

(underline: primary key,
bold: foreign key)

user (uid, user_name, pick_clothes)
machine (mid, bid, **quid**, time_stamp)
laundry_bag (**quid**, **uid**, type)
cloth (cid, **uid**, type, status)
query (quid, **uid**, query_id, status, time_stamp, type)
booking_query (bquid, **uid**, num, **bid**, time_stamp)
add_remove_user_query (uquid, uid, user_name, add_remove, time_stamp)
employee (eid, name, **bid**)
branch (bid, bname, bcity)

c. Data normalization.

1NF: First normal form

There are no multi-valued attributes. That is, every table cell has only one value and all tuples are unique.

2NF: Second normal form

Table is in first normal form.
No key is determined by the proper subset of a candidate key.

3NF: Third normal form

Table is in second normal form.
There are no transitive dependency.

All tables in the Laundromat system have been created with these rules in mind. Hence, all of them are in 3NF.

d. List of tables required.

- user
- machine
- laundry_bag
- cloth
- query
- booking_query
- add_remove_user_query
- employee
- branch

e. Additional components
(by order of appearance in SQL query file)

1. procedure machine_Available(in bid1 int, in number_clothes int, out check1 bit): given the number of clothes, this procedure checks if there are enough machines available to wash them by the branch id.
2. procedure machine_Available_by_city(in city_name varchar(50), in number_clothes int, out check1 bit): given the number of clothes, this procedure checks if there are enough machines available to wash them by the branch city.
3. procedure register_user_query(in username varchar(50), in uid1 int, out sta bit): tthis procedure is used by the user to send a register request which may or may not be processed by the admin.
4. procedure register_user(in quid1 int, out sta bit): the admin, by reading the quid from query table regarding a user register query, can register that user. This fails if the user already exists.
5. procedure deregister_user_query(in uid1 int): sent by user to request admin to deregister itself.
6. procedure deregister_user(in quid1 int, out sta bit): procedure used by admin to approve deregister request of the user. It fails if the user does not already exist.
7. procedure send_book_query(in uid int (11), in num int(11), in branch varchar(50)): sent by the user to request admin to add num number of clothes at the branch the user wants to.
8. procedure add_cloth(uid1 int(11), type1 varchar(50), quid1 int(11)): user adds clothes to his/her laundry bag using this procedure as per his booking query.
9. procedure book(in quid1 int(11), out out1 bit): the admin, if adequate machines are available, set them to work and clothes are automatically shifted from laundry bag of the user to the firm's cloth table.

API- backend

All the backend is hosted on this link(base-address)
https://laundrymanage.herokuapp.com/
(you will see it as blank because it's backend base address)
Options available-
Adding User-
https://laundrymanage.herokuapp.com/adduser?name=raj&id=10021
Removing User-
https://laundrymanage.herokuapp.com/removeuser?id=10021
Updating Machines-
https://laundrymanage.herokuapp.com/updatemachines
Getting whole list of queries-
https://laundrymanage.herokuapp.com/getQueries
Getting Pending queries-
https://laundrymanage.herokuapp.com/getPendingQueries

Accepting a query-
https://laundrymanage.herokuapp.com/acceptQueries?quid=694
Machine available query by user-
https://laundrymanage.herokuapp.com/machineavailable?bid=1&num_clothes=12
https://laundrymanage.herokuapp.com/machineavailablebycity?bcity=Pilani&num_clothes=12
Getting list of all users-
https://laundrymanage.herokuapp.com/getallusers
Sending booking query-
https://laundrymanage.herokuapp.com/sendbookquery?uid=10021&num=2&bname=Delhibranch
Checking status of the query-
https://laundrymanage.herokuapp.com/checkstatusquery?uid=694
Retrieving clothes-
https://laundrymanage.herokuapp.com/retrive?quid=694&uid=10021
Adding clothes-
https://laundrymanage.herokuapp.com/add_clothes?uid=10021&type=socks&quid=694
Getting List of Machines-
https://laundrymanage.herokuapp.com/get_machines

For backend we have used node.js+express.js
And for rendering frontend we have used ejs
You can checkout the frontend at
http://laundrymanaging.herokuapp.com/machines