CS2102 Project Report

Project Topic: Topic D Carpooling

**Group 20**

Justin Poh

Conant Teo

Jonathan Tan A0156136H

Lee Wei Qing

# 1. Server

# 2. ER Diagram

# 3. Database Schema

## Stored procedure to add a user

CREATE OR REPLACE FUNCTION add\_user(\_name varchar, \_gender varchar, \_contact varchar, \_email varchar, \_password varchar, \_platenumber varchar, \_capacity integer, \_isDriver boolean, \_isAdmin boolean)

returns varchar as

$BODY$

BEGIN

IF EXISTS (SELECT email FROM useraccount WHERE email = \_email) THEN return 'Email already exists!';

END IF;

IF EXISTS (SELECT vehicle\_plate FROM useraccount WHERE vehicle\_plate = \_platenumber) THEN return 'Plate number already exists!';

END IF;

INSERT INTO useraccount VALUES (\_name, \_gender, \_contact, \_email, \_password, \_platenumber, \_capacity, \_isDriver, \_isAdmin);

return 'Account has been successfully created!';

END;

$BODY$

language 'plpgsql' volatile;

Description: Performs constraint checks and add user into the useraccount table in the database.

## Stored procedure to add an advertisement

CREATE OR REPLACE FUNCTION add\_advertisement(\_email varchar, \_startLocation varchar, \_endLocation varchar, \_pickupDate date, \_pickupTime time, \_selfSelect boolean)

returns varchar as

$BODY$

BEGIN

IF NOT EXISTS (SELECT email FROM useraccount WHERE email = \_email)

THEN RETURN 'Account for this email does not exist!';

ELSEIF (\_startLocation = \_endLocation)

THEN RETURN 'Cannot have the same start and end location!';

ELSEIF ((\_pickupDate + \_pickupTime) <= (current\_timestamp + INTERVAL '1 hour'))

THEN RETURN 'Pick-up date and time must be at least 1 hour from now!';

END IF;

INSERT INTO advertisements(email\_of\_driver, start\_location, end\_location, creation\_date\_and\_time, date\_of\_pickup, time\_of\_pickup, self\_select)

VALUES (\_email,\_startLocation, \_endLocation, current\_timestamp, \_pickupDate, \_pickupTime, \_selfSelect);

RETURN '';

END;

$BODY$

language 'plpgsql' volatile;

Description: Performs constraint checks and adds an advertisement into the advertisements table in the database.

## Stored trigger to update bids and advertisements

CREATE TRIGGER updateBidsAndOffer

AFTER UPDATE

ON bid

FOR EACH ROW

EXECUTE PROCEDURE updateBidsAndOfferStatus();

CREATE OR REPLACE FUNCTION updateBidsAndOfferStatus()

RETURNS TRIGGER AS $bid\_table$

BEGIN

IF NEW.status = 'Accepted'

THEN

UPDATE bid SET status = 'Rejected' WHERE status = 'Pending' AND advertisementID = NEW.advertisementID;

INSERT INTO bidHistory(email, status, price, creation\_date\_and\_time, start\_location, end\_location, date\_of\_pickup, time\_of\_pickup)

SELECT B1.email, B1.status, B1.price, B1.creation\_date\_and\_time, A1.start\_location, A1.end\_location, A1.date\_of\_pickup, A1.time\_of\_pickup FROM bid B1, advertisements A1 WHERE B1.advertisementID = A1.advertisementID AND A1.advertisementID = NEW.advertisementID;

DELETE FROM bid WHERE advertisementID = NEW.advertisementID;

UPDATE advertisements SET closed = true WHERE advertisementID = NEW.advertisementID;

INSERT INTO advertisementsHistory(email\_of\_driver, start\_location, end\_location, creation\_date\_and\_time, date\_of\_pickup, time\_of\_pickup, self\_select)

SELECT email\_of\_driver, start\_location, end\_location, creation\_date\_and\_time, date\_of\_pickup, time\_of\_pickup, self\_select FROM advertisements WHERE advertisementID = NEW.advertisementID;

DELETE FROM advertisements WHERE advertisementID = NEW.advertisementID;

ELSEIF NEW.status = 'Rejected' OR NEW.status = 'Offer retracted' OR NEW.status = 'Offer expired' THEN

INSERT INTO bidHistory(email, status, price, creation\_date\_and\_time, start\_location, end\_location, date\_of\_pickup, time\_of\_pickup)

SELECT B1.email, B1.status, B1.price, B1.creation\_date\_and\_time, A1.start\_location, A1.end\_location, A1.date\_of\_pickup, A1.time\_of\_pickup FROM bid B1, advertisements A1 WHERE B1.advertisementID = A1.advertisementID AND A1.advertisementID = NEW.advertisementID;

DELETE FROM bid WHERE advertisementID = NEW.advertisementID AND email = NEW.email;

END IF;

RETURN NULL;

END;

$bid\_table$ LANGUAGE plpgsql;

Description: Triggers updateBidsAndOfferStatus() whenever the advertisements table is updated.

If a bid has its status turned to “Accepted”, then the other bids for the same advertisement will have their status turned into “Rejected”, and all these bids will be removed from the bid table and inserted into the bidHistory table, while the corresponding advertisement will be removed from the advertisements table and inserted into the advertisementHistory table.

If a bid has its status turned to ‘Rejected’, ‘Offer retracted’, ‘Offer expired’

# 4. Web app screenshots