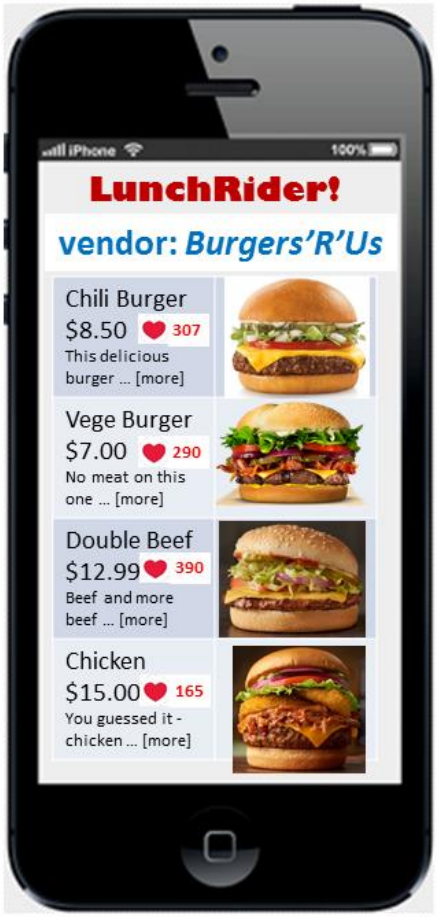
Assignment One – Data Modelling

INFO90002 - 2018 - sem1

LunchRider

You are designing a database to support *LunchRider*, a new startup in the food delivery business. *LunchRider* allows people to order lunches from local food vendors and have them delivered by a delivery rider on a bike. *LunchRider* is ambitious and plans to eventually have 20 million customers, buying meals from 10,000 vendors, delivered by 20,000 riders.



Not all tables and columns are necessarily described explicitly in this document: you will need to consider carefully how the system works in order to deduce the complete data model.

How *LunchRider* works

When a customer opens the *LunchRider* app, it first presents a list of local vendors such as cafes, restaurants and snack bars. The customer clicks on a vendor. Then the app shows the meals available from that cafe.

The customer chooses which meals they want from that vendor: for example "2 chili burgers and 1 mango smoothie". The phone sends this order to the LunchRider server, along with the phone's GPS coordinates - the meals will be delivered to this location.

Customers can click to "like" a meal. (Later they can undo their like via an "unlike" button.) The total number of likes (across all customers) is displayed beside each meal.

When a customer's order is received, *LunchRider* broadcasts a work offer to riders who are near the customer. Riders see the offer pop up on their app, and can press "accept" or "no thanks". If several riders press "accept", our algorithm chooses one rider and gives them the job (based on the rider's ratings.) If no rider accepts the offer within a few minutes, another offer is broadcast, and so on, until a rider has taken the job. We need to keep track of these offers and responses.

The chosen rider now goes to the vendor, picks up the meals, and delivers them to the customer's location. We record at what time the rider delivers the order. Payment is automatically deducted from the customer's credit card: we don't need to record payments – this is handled by the bank.

After delivery, the app allows the customer to rate the rider's service, choosing from 1 star (worst), 2, 3, 4 or 5 stars (best). The app also allows the customer to add the rider to their "favourite" or "don't send again" list. *LunchRider* uses this information to help choose riders for future work offers.

People

If someone wants to order a meal, they need to first register as a customer, giving *LunchRider* their given and family names, email address, and mobile phone number.

given name	
family name	
email	
phone	
credit card	

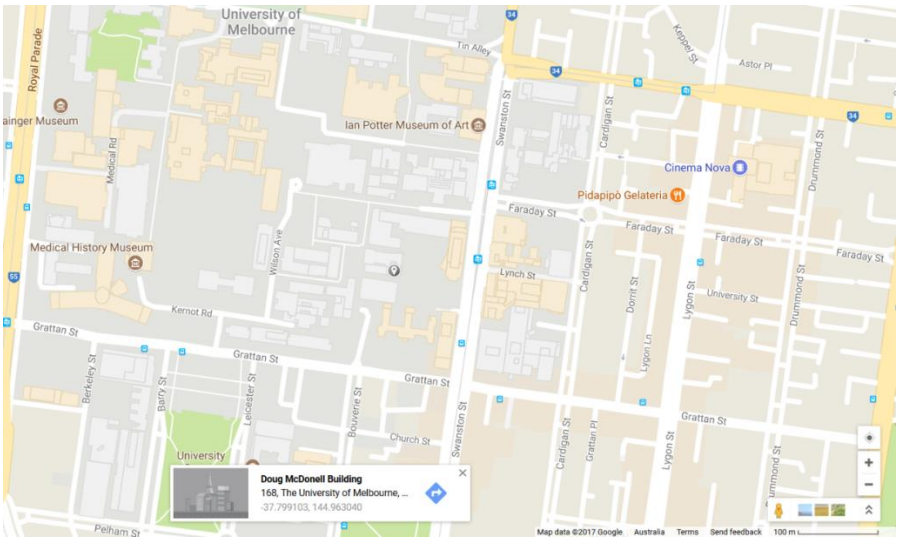
They must also register a credit card. When they buy a meal, the payment is automatically deducted from this card. Over time a customer may register more than one credit card: we keep track of them all, but must record which is their "current" card for payment. About each card we record its 16-digit number and month/year of expiry.

Vendors must register the name and address of their business, including GPS coordinates and a contact email address, and provide the name, price, description (max 1,000 characters) and photo of each meal that they want to sell. We will store photos in the database.

If someone wants to be a delivery rider, they need to first register, giving *LunchRider* their first and last name, mobile phone number, and date of birth. Whenever a rider is on duty, the rider's app sends us their GPS coordinates about once per minute, allowing us to keep track of each rider's location. We must store a history of all the rider locations we receive, for later analysis.

Location coordinates

All locations are recorded as a pair of numbers representing latitude and longitude. Latitudes are between -90 and 90 degrees (south pole to north pole) while longitudes are between -180 and 180 degrees (west or east of the prime meridian). *LunchRider* uses a precision of 4 decimal places, which is about 11 metres at the equator (smaller in Melbourne). For example, the Doug McDonell building at The University of Melbourne is at latitude -37.7989, longitude 144.9627.



[end of assignment]

