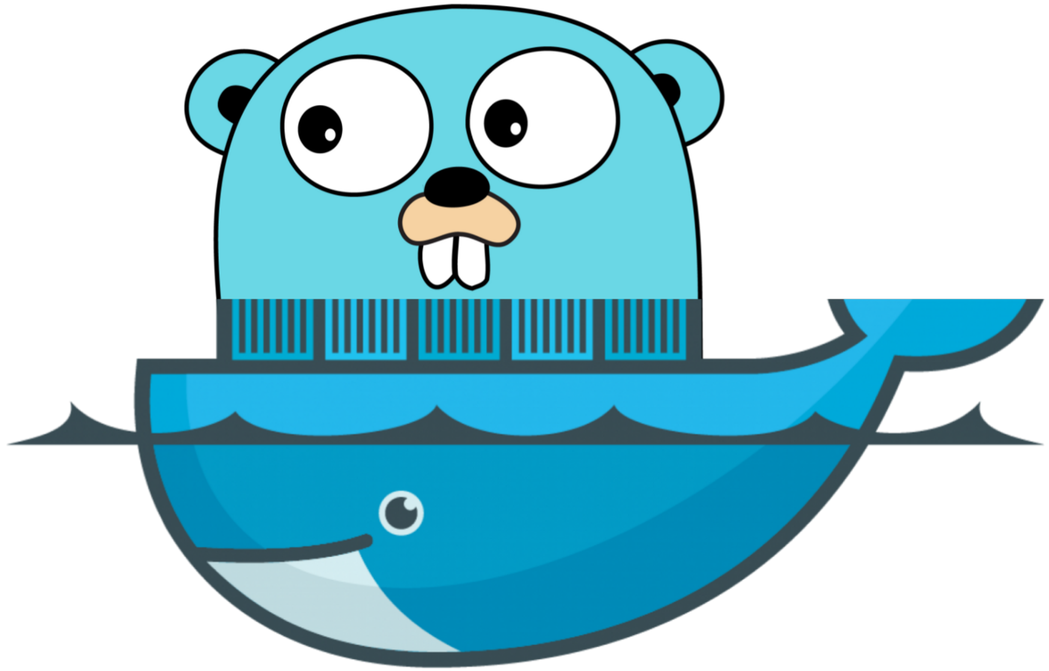
Golang Engineer Training

Analyze UI to Database schema



200Lab Education

Designed by **200lab Education - Nâng tầm chuyên môn, định hướng tương lai**
Mentor/Instructor: Viet Tran - Solution Architect 200lab

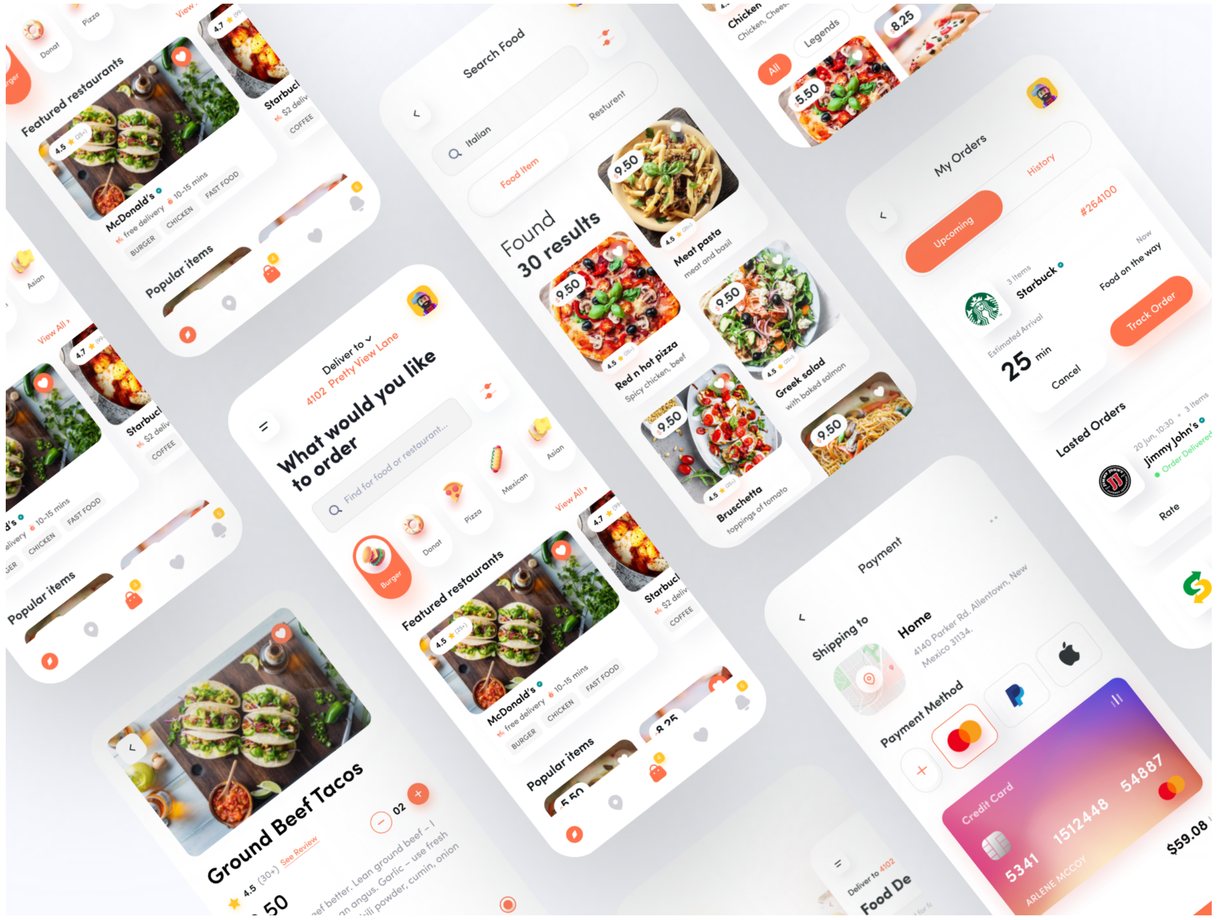
200Lab Education

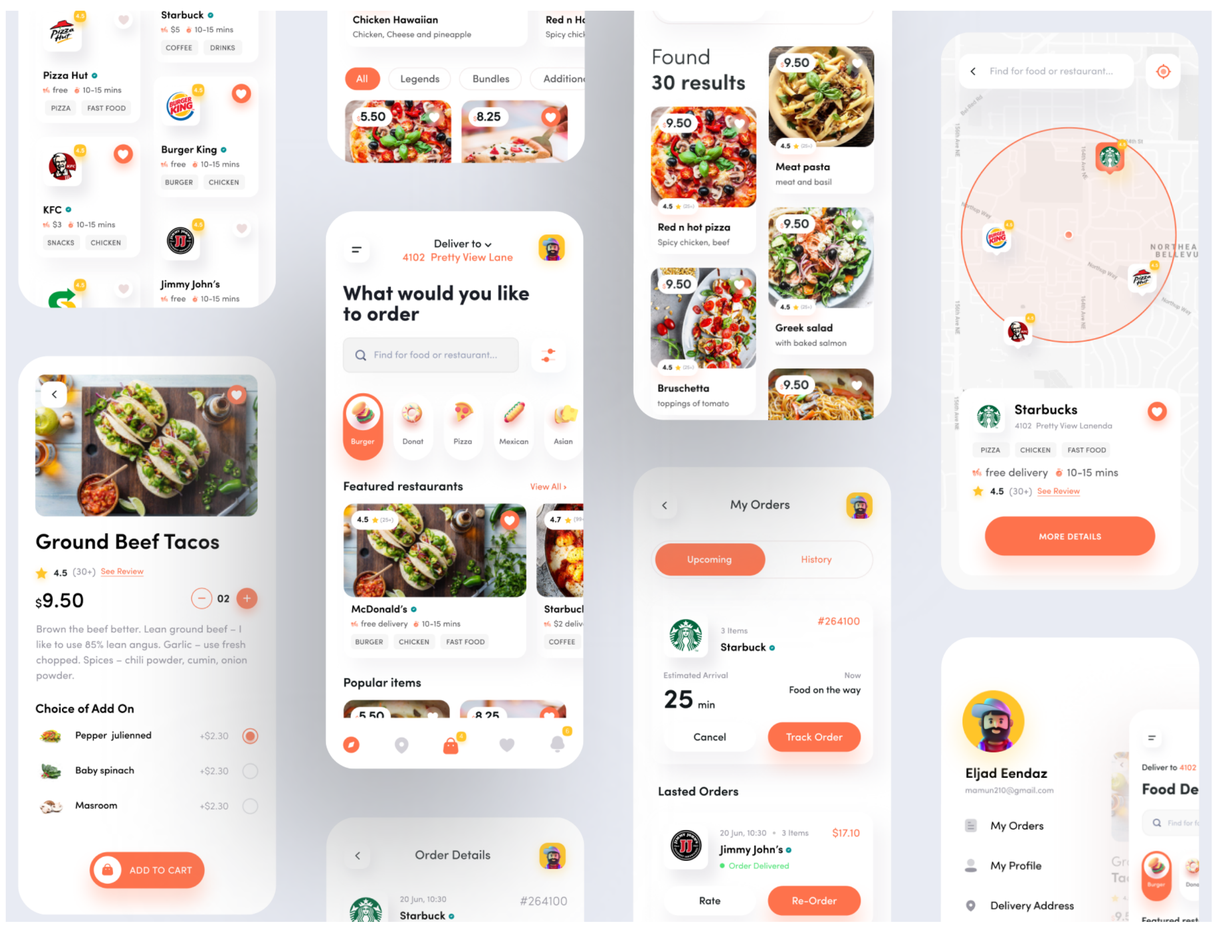
Agenda

- Analyze UI Food Delivery (from 200Lab)
- UI to User Story / Requirement (Tech)
- User Story to Database (Entity and Relationship) or ERD
- Bonus:
  - Improve DB, high performance.
  - Multi-column indexes in right way.
  - We do not use foreign key any more.

200Lab Education

UI for example





https://ui8.net/prelook-studio/products/food-delivery-ui-kit-foodhub

(*) Everyone (200Lab's student) will receive this UI (design files).

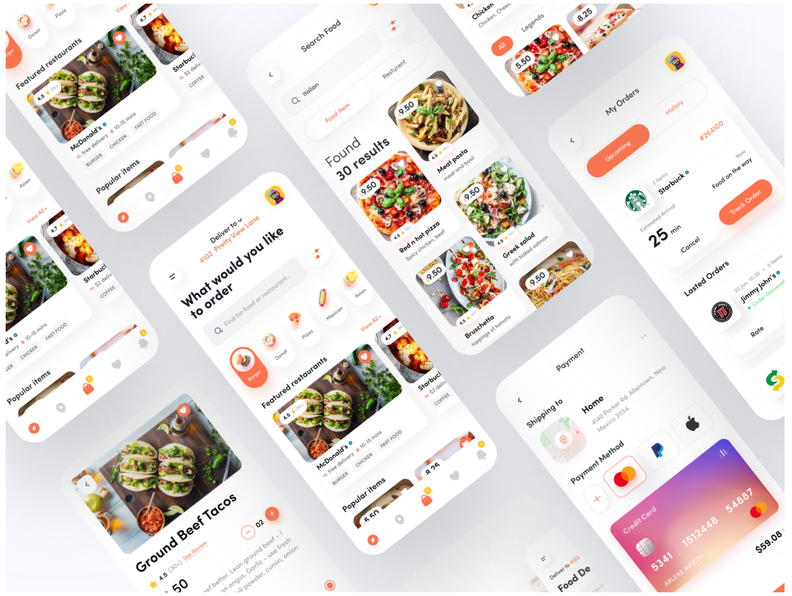
200Lab Education

Why from UI?
Not from requirement!?

200Lab Education

UI to User Story (exp from 200Lab)

- What's kind/category of the app?
- How many roles of user?
- Focus on flow of screens (if available).
- Check UI elements carefully, some of them might have special features.
- Demo on UI (200Lab)



200Lab Education

User story/feature to Model/Entity

- Focus on "noun" in user story. Ex:
  - As an user, I can browse all restaurants.
  - As an user, I can browse all foods in restaurant.
- So we have: user, restaurant and food are entities.
- Note:
  - Entity name should be in **singular** form.
  - Table name should be in **plural** form.
  - High priority on important/core features.

200Lab Education

User story/feature to Model/Entity (.cont)

- Entity properties:
  - Some of them can be found on list UI.
  - More on details screens.
- Note:
  - Not all of element on UI are properties. They could be computed fields.
  - Each of them should have ID, status, created_at and updated_at.

200Lab Education

User story/feature to Model/Entity (.cont)

- Let’s say Food entity:
  - **id** is int (primary key, auto increasement).
  - **name** is varchar (string in MySQL).
  - **restaurant_id** is int (should be not null).
  - **price** is float
  - **image** is json (storing object as JSON string)
  - **status** is int (or enum('active', 'inactive')).
  - **created_at** is timestamp (default is now()).
  - **updated_at** is timestamp (default is now(), auto change).

200Lab Education

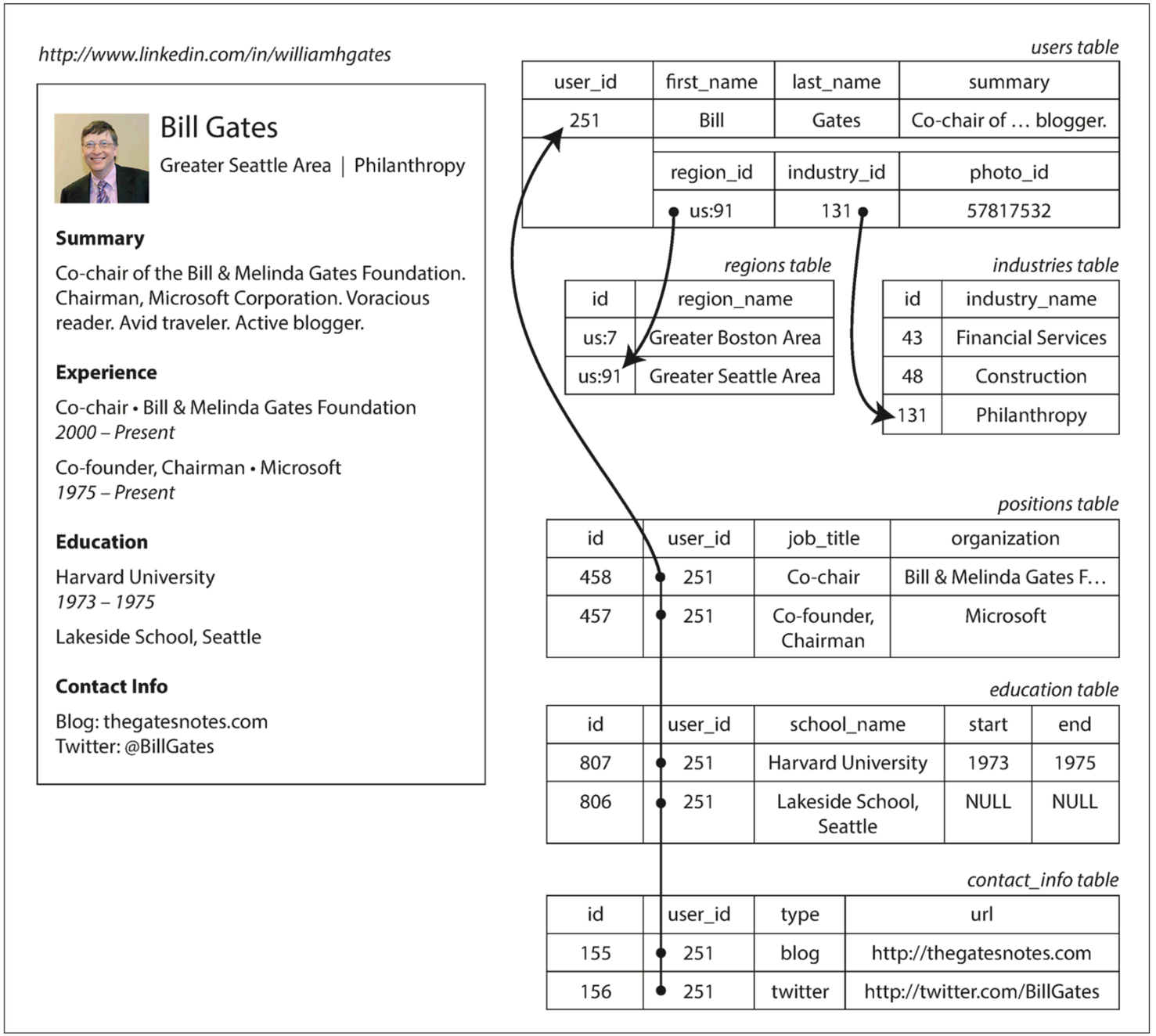
User story/feature to Model/Entity (.cont)

- But these are not properties:
  - **has_liked** is tinyint(1) (bool in MySQL).
  - **liked_count** is int.
  - **rating_point** is float.
- Note:
  - Some of them can be computed, not stored.
  - Sometimes we choose storing them for caching (speed up reading but slow down updating).

200Lab Education

Entity Relationship (explanation)

- One to one
- One to many
- Many to many



DB Designing (demo from Twitter)

Source: "Designing data-intensive applications" ebook

200Lab Education

Install & Run MySQL (with Docker)

```
docker run -d --name mysql  --privileged=true -e
MYSQL_ROOT_PASSWORD="ead8686ba57479778a76e"   -e
MYSQL_USER="food_delivery"   -e
MYSQL_PASSWORD="19e5a718a54a9fe0559dfbce6908"   -e
MYSQL_DATABASE="food_delivery"  -p 3306:3306   bitnami/mysql:5.7
```

- Root Password: **ead8686ba57479778a76e** ("root" user only)
- User: **food_delivery**
- User pass: **19e5a718a54a9fe0559dfbce6908**
- DB Name: **food_delivery**
- Port: **3306**

200Lab Education

Connecting to MySQL (with UI Tools)

- Navicat (https://www.navicat.com/en/)
- Table Plus (https://tableplus.com) (*)
- PHPMyAdmin (include in XAMMP)

(*) It’s FREE and simple to use. Developed by a Vietnamese developer.

200Lab Education

Demo with TablePlus (and learn SQL syntax)

200Lab Education

Do it yourself

- Write down all user stories in UI Food Delivery.
- Design all entities and their relationships (your way).
- Hint: use https://dbdiagram.io for design your DB.

200Lab will **review** and give you some **advice** you when you finish! Try your best!!

200Lab Education

Thank you.