

# ORDER-N-GET

## ONLINE DISH ORDER SYSTEM

YANG, XING	010696101
JIN, WEIYU	011819756
WANG, WENQI	012491076
LI, HONGYUAN	011838151
WANG, XIAOMAN	007840534

# CONTENT

1.Introduction

2.Architecture

3.Transactions

3.Database Design

4.Triggers & Stored Procedures

6.Function & Demo

7.Retrospective

# INTRODUCTION

Problem:

Restaurant:

- Want to make more money but the seat is limited.
- High cost on hire a new employee to do delivery service.
- Want to use the internet to advertise their restaurant.

Customer:

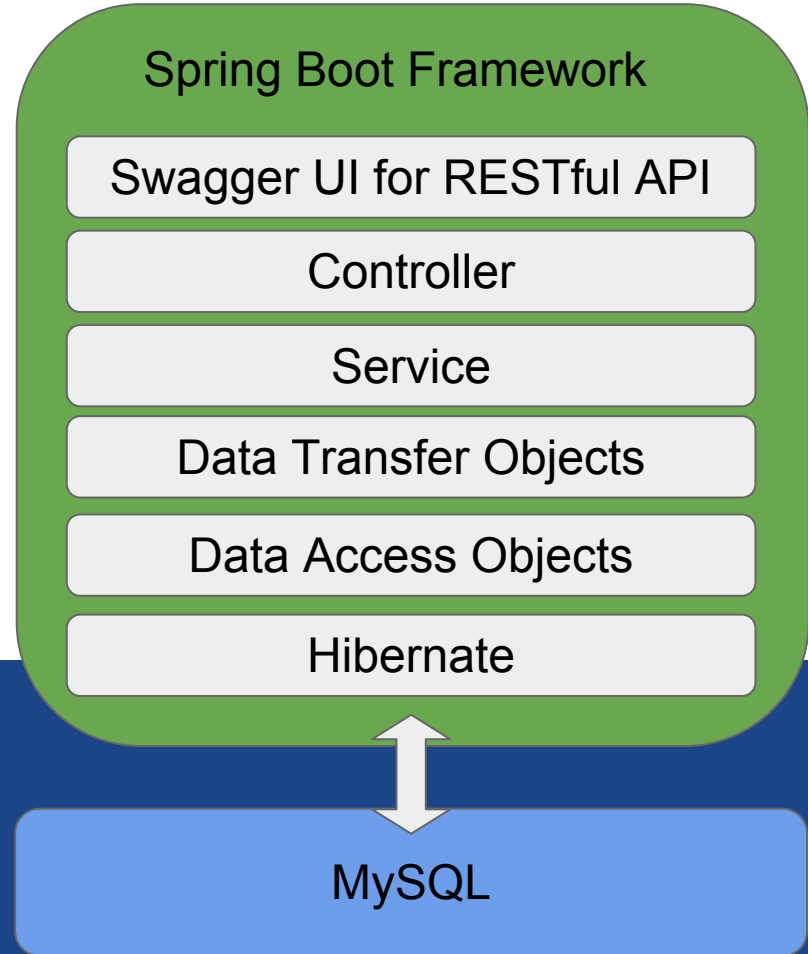
- Some restaurant's waiting time is too long.
- Want to see restaurants information in their area.
- Want delivery service.

Solution:

Order-N-Get Online Dish Order System



# ARCHITECTURE



# TRANSACTIONS

Every Controller Method (e.g. GET, PUT, POST, and DELETE) is a Transaction

- Flexible Annotation
- Complex Transaction Logic
- Easy to Manage

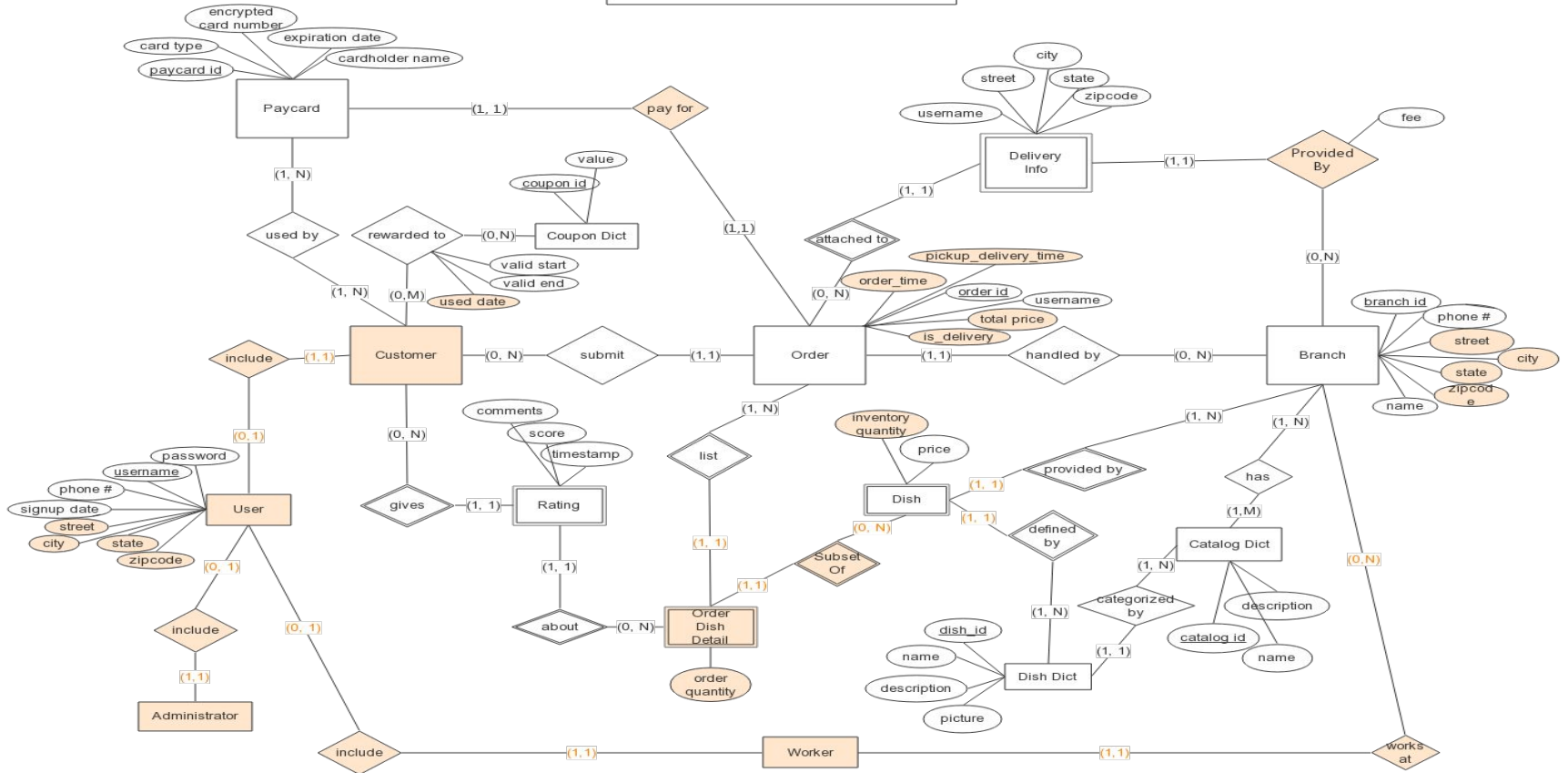
```
@EnableTransactionManagement
// Add before Main App Class

new HibernateTransactionManager();
// Add to Hibernate Config

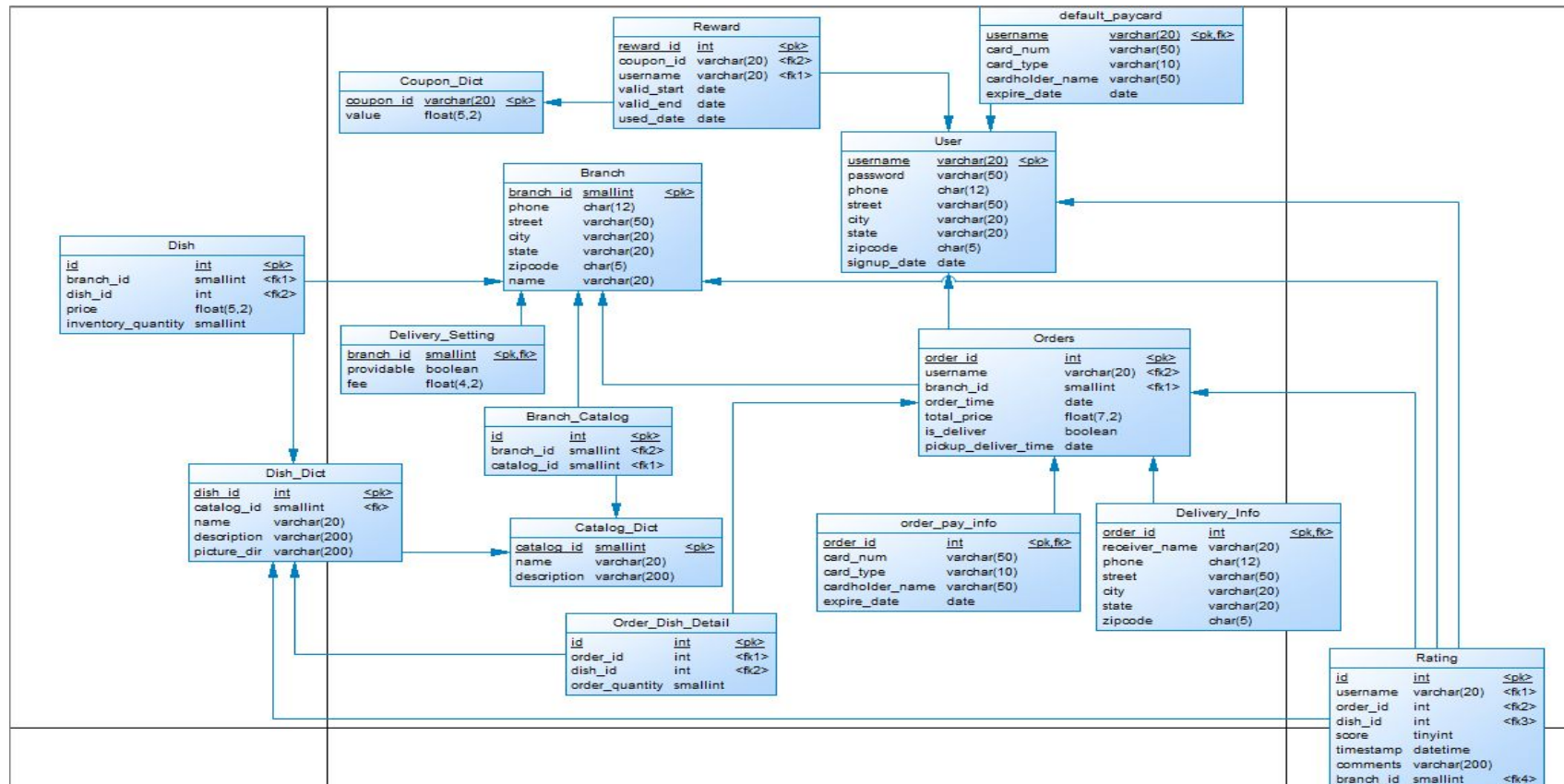
@Transactional(rollbackFor =
Exception.class)
// Add before Controller Class
```

# DATABASE DESIGN

Team 3: Dish Ordering System



# DATABASE DESIGN



# TRIGGERS

```
DELIMITER $$
CREATE TRIGGER send_commitReward AFTER INSERT ON Rating
FOR EACH ROW

BEGIN
    DECLARE rewards Integer;

    SELECT count(*) INTO rewards FROM RATING WHERE order_id = NEW.order_id;

    IF ( rewards = 1) THEN

        INSERT INTO REWARD (coupon_id, username, valid_start,valid_end) VALUES
        ( 'commentReward',NEW.username, now(), (now() + INTERVAL 20 DAY));

    END IF;

END $$
DELIMITER ;
```



# STORED PROCEDURE

The screenshot shows a database management interface with a left sidebar for 'SCHEMAS' and a main editor for SQL code. The sidebar lists a 'demo' schema and a 'dos' schema. Under 'dos', there are 'Tables', 'Views', 'Stored Procedures' (containing 'autoConfirm'), and 'Functions'. The main editor displays SQL code for creating a stored procedure and an event. The code is as follows:

```
352 DELIMITER $$
353
354 CREATE PROCEDURE dos.autoConfirm()
355 BEGIN
356
357     update orders
358     set pickup_deliver_time = now()
359     where DATE_SUB(NOW(), INTERVAL 20 DAY) > order_time and PICKUP_DELIVER_TIME is null;
360
361 END $$
362 DELIMITER ;
363
364
365
366 DELIMITER $$
367 CREATE EVENT myevent
368 ON SCHEDULE EVERY 5 DAY
369 STARTS CURRENT_TIMESTAMP()
370
371 DO
372 call dos.autoConfirm();
373 $$
374 DELIMITER ;
375
```

The interface includes a search bar for 'Filter objects', a toolbar with various icons, and a 'Limit to 1000 rows' dropdown menu.

DEMO TIME!

# RETROSPECTIVE

What we did well?

- Develop as a team
- Learn from practice

What we learned?

- Database service.
- Spring Framework.
- TeamWork Communication.
- Git management.

What can be improved?

- Do it after communication
- More function(inventory management, more analysis function based on data)
- Front end

THANK YOU!