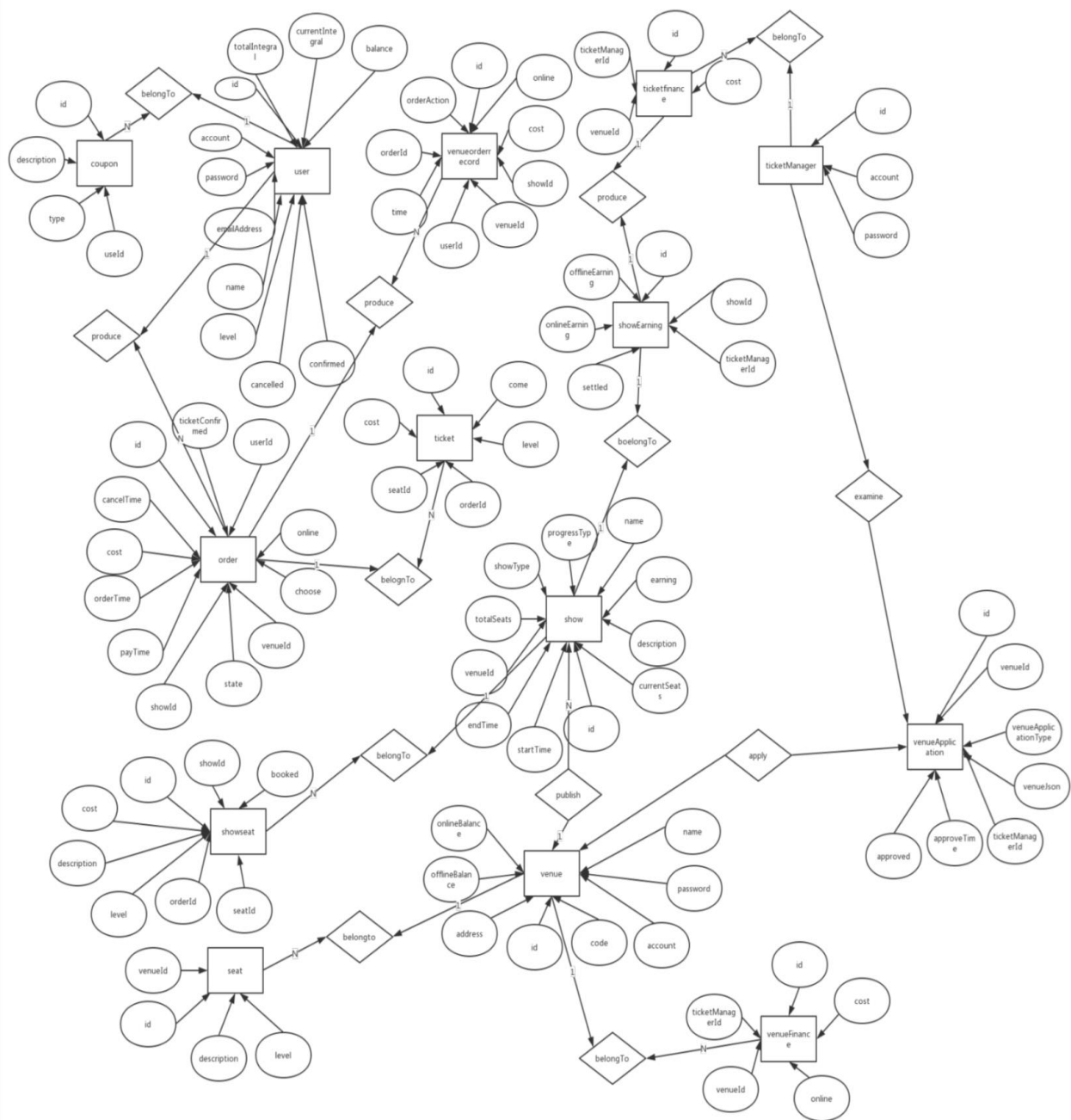


## 一. 数据库设计

### 1. ER 图



### 2. 数据库表

```

CREATE TABLE coupon
(
  id INT NOT NULL
    PRIMARY KEY,
  description VARCHAR(255) NULL,
  type INT NOT NULL,
  used BIT NOT NULL,
  userId INT NULL
)
ENGINE = MyISAM;

CREATE INDEX FK6f3wdb53ors1b5s8x21ea6slo
ON coupon (userId);

```

```

CREATE TABLE ordertable
(
  id INT AUTO_INCREMENT
    PRIMARY KEY,
  cancelTime VARCHAR(255) NULL,
  cost DOUBLE NOT NULL,
  orderTime VARCHAR(255) NULL,
  payTime VARCHAR(255) NULL,
  showId INT NOT NULL,
  state VARCHAR(255) NULL,
  ticketConfirmedTime VARCHAR(255) NULL,
  userId INT NOT NULL,
  venueId INT NOT NULL,
  choose BIT NOT NULL,
  online BIT NOT NULL
)
ENGINE = MyISAM;

```

```

CREATE TABLE seat
(
  id INT AUTO_INCREMENT
    PRIMARY KEY,
  description VARCHAR(255) NULL,
  level VARCHAR(255) NULL,
  venueId INT NULL
)
ENGINE = MyISAM;

CREATE INDEX FKhekuib7o6c87pqfs1upd841va
ON seat (venueId);

```

```
CREATE TABLE showtable
(
  id INT AUTO_INCREMENT
    PRIMARY KEY,
  currentSeats INT NOT NULL,
  description VARCHAR(255) NULL,
  earning DOUBLE NOT NULL,
  name VARCHAR(255) NULL,
  progressType VARCHAR(255) NULL,
  showType VARCHAR(255) NULL,
  totalSeats INT NOT NULL,
  venueId INT NOT NULL,
  endTime VARCHAR(255) NULL,
  startTime VARCHAR(255) NULL
)
ENGINE = MyISAM;
```

```
CREATE TABLE showearning
(
  id INT NOT NULL
    PRIMARY KEY,
  offlineEarning DOUBLE NOT NULL,
  onlineEarning DOUBLE NOT NULL,
  settled BIT NOT NULL,
  showId INT NOT NULL,
  ticketManagerId INT NOT NULL
)
ENGINE = MyISAM;
```

```
CREATE TABLE showseat
(
  id INT AUTO_INCREMENT
    PRIMARY KEY,
  cost DOUBLE NOT NULL,
  description VARCHAR(255) NULL,
  level VARCHAR(255) NULL,
  orderId INT DEFAULT '-1' NULL,
  seatId INT NOT NULL,
  showId INT NULL,
  booked BIT NOT NULL
)
ENGINE = MyISAM;

CREATE INDEX FKIr4q3na77np5h3ie0sqm9atlh
ON showseat (showId);
```

```
CREATE TABLE ticket
(
  id          INT AUTO_INCREMENT
    PRIMARY KEY,
  cost        DOUBLE          NOT NULL,
  seatId      INT             NULL,
  orderId     INT             NULL,
  level       VARCHAR(255)    NULL,
  come        BIT             NOT NULL
)
ENGINE = MyISAM;

CREATE INDEX FKfg57j6f09dio5g2m5yhy6cx7h
ON ticket (orderId);
```

```
CREATE TABLE ticketfinance
(
  id          INT NOT NULL
    PRIMARY KEY,
  cost        DOUBLE NOT NULL,
  ticketManagerId INT NOT NULL,
  venueId     INT NOT NULL
)
ENGINE = MyISAM;
```

```
CREATE TABLE ticketmanager
(
  id          INT AUTO_INCREMENT
    PRIMARY KEY,
  account     VARCHAR(255) NOT NULL,
  password    VARCHAR(255) NOT NULL
)
ENGINE = InnoDB;
```

```
CREATE TABLE user
(
  id          INT AUTO_INCREMENT
    PRIMARY KEY,
  account     VARCHAR(255) NOT NULL,
  password    VARCHAR(255) NOT NULL,
  emailAddress VARCHAR(255) NULL,
  name        VARCHAR(255) NULL,
  level       VARCHAR(255) NULL,
  totalIntegral INT DEFAULT '0' NULL,
  cancelled   TINYINT(1) DEFAULT '0' NOT NULL,
  confirmed   TINYINT(1) DEFAULT '0' NOT NULL,
  currentIntegral INT NOT NULL,
  balance     DOUBLE NOT NULL
)
ENGINE = InnoDB;
```

```
CREATE TABLE venue
(
  id INT AUTO_INCREMENT
    PRIMARY KEY,
  code VARCHAR(255) NULL,
  account VARCHAR(255) NULL,
  password VARCHAR(255) NULL,
  name VARCHAR(255) NULL,
  address VARCHAR(255) NULL,
  offlineBalance DOUBLE NOT NULL,
  onlineBalance DOUBLE NOT NULL
)
ENGINE = InnoDB;
```

```
CREATE TABLE venueapplication
(
  id INT AUTO_INCREMENT
    PRIMARY KEY,
  venueId INT NULL,
  venueApplicationType VARCHAR(255) NULL,
  venueJson TEXT NULL,
  ticketManagerId INT NULL,
  approveTime VARCHAR(255) NULL,
  approved BIT NOT NULL
)
ENGINE = InnoDB;
```

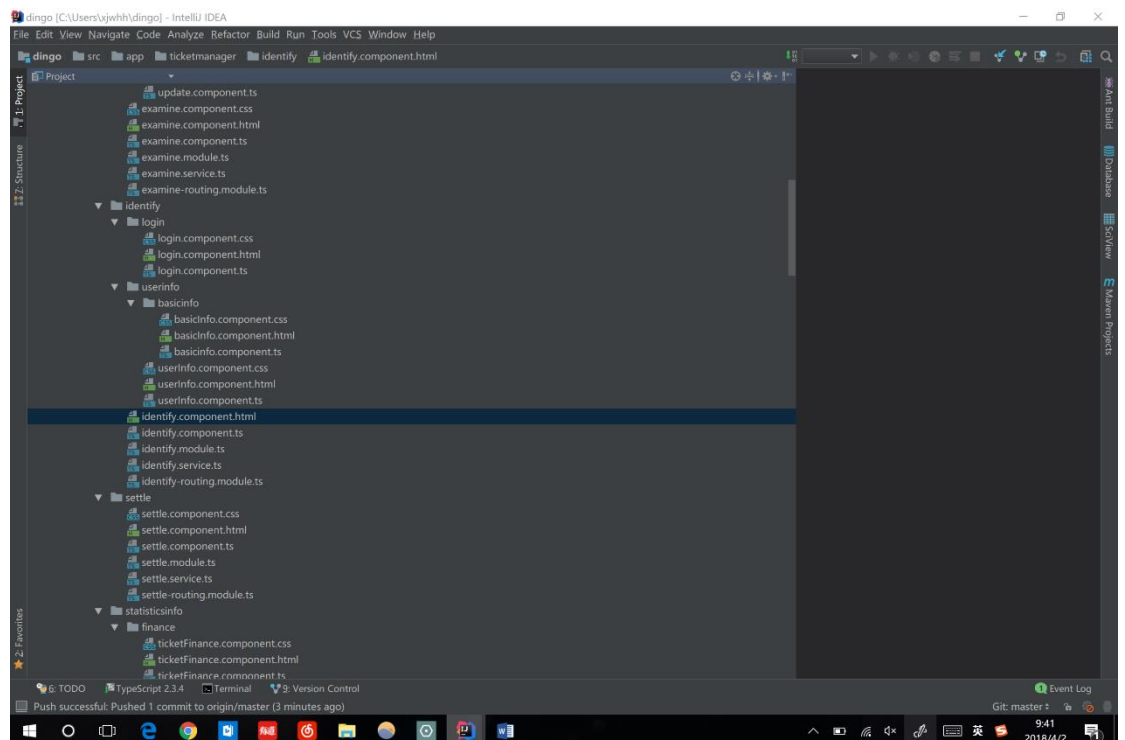
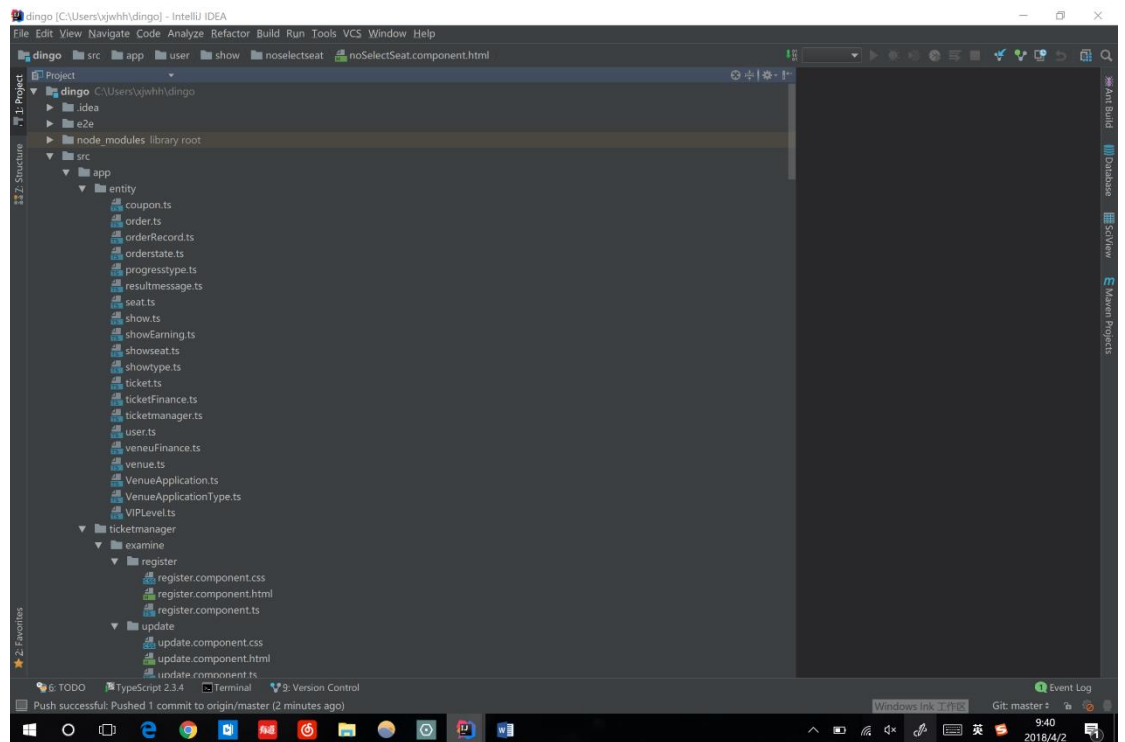
```
CREATE TABLE venuefinance
(
  id INT AUTO_INCREMENT
    PRIMARY KEY,
  cost DOUBLE NOT NULL,
  ticketManagerId INT NOT NULL,
  venueId INT NOT NULL,
  onLine BIT NOT NULL
)
ENGINE = MyISAM;
```

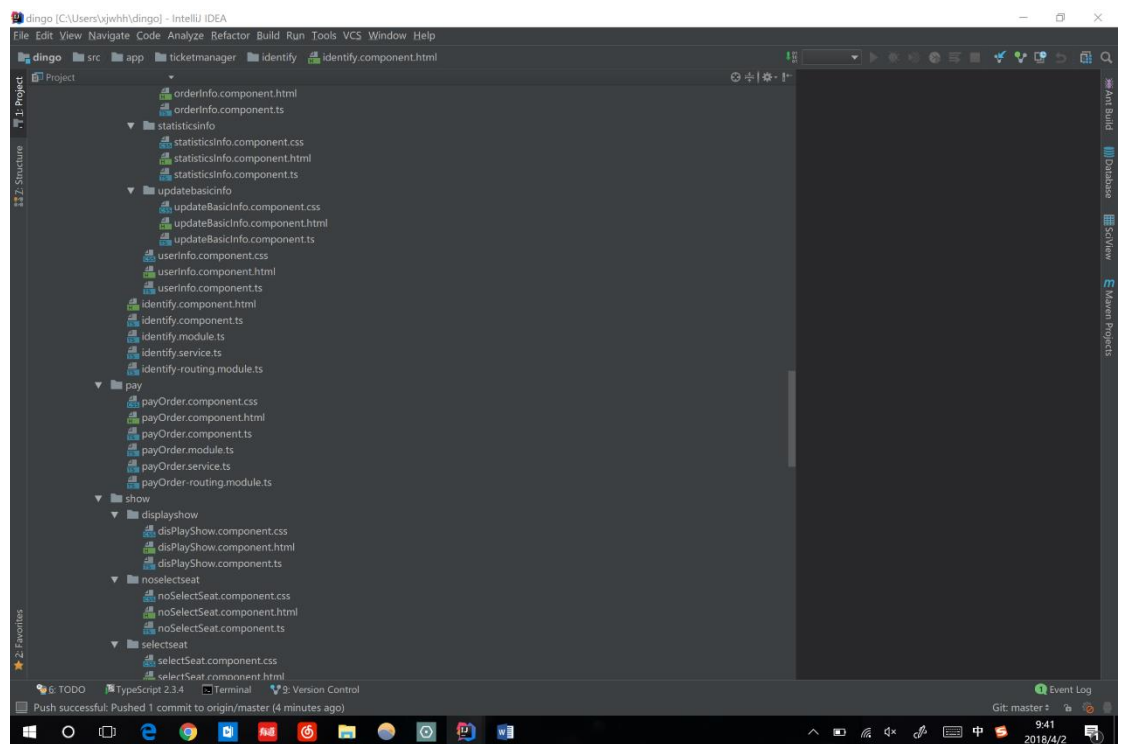
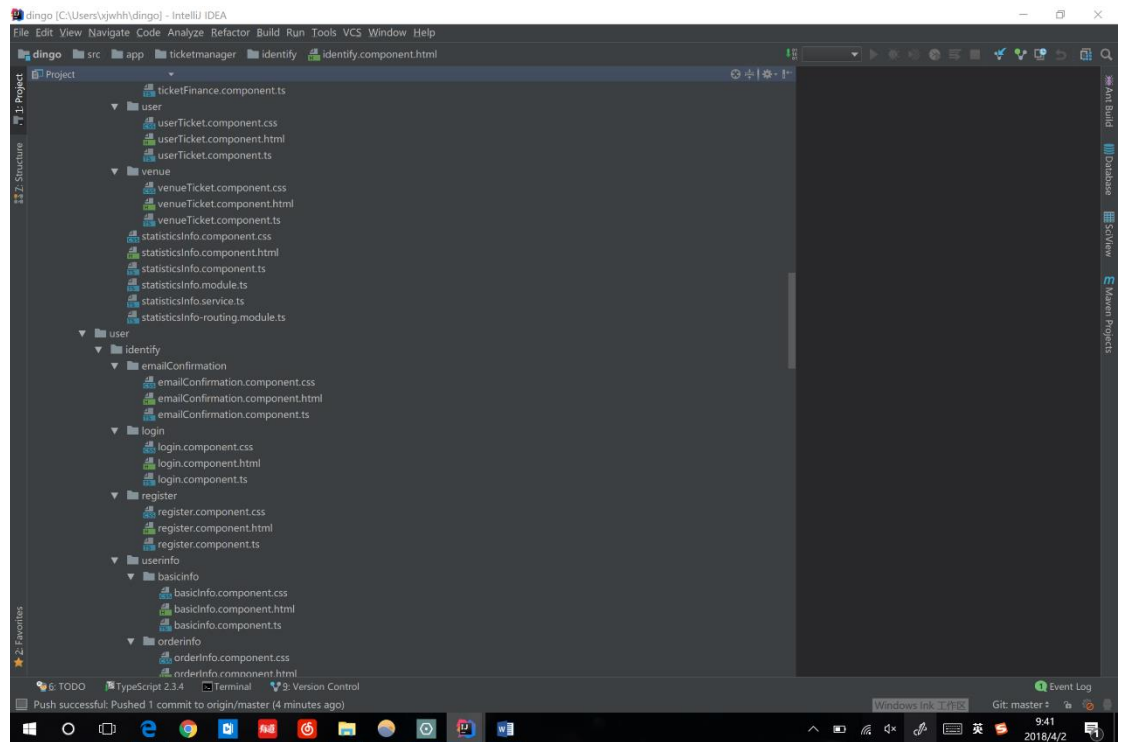
```
CREATE TABLE venueorderrecord
(
  id INT AUTO_INCREMENT
    PRIMARY KEY,
  orderAction VARCHAR(255) NULL,
  orderId INT NOT NULL,
  time VARCHAR(255) NULL,
  userId INT NOT NULL,
  venueId INT NOT NULL,
  showId INT NOT NULL,
  cost DOUBLE NOT NULL,
  online BIT NOT NULL
)
ENGINE = MyISAM;
```

## 二. 架构设计

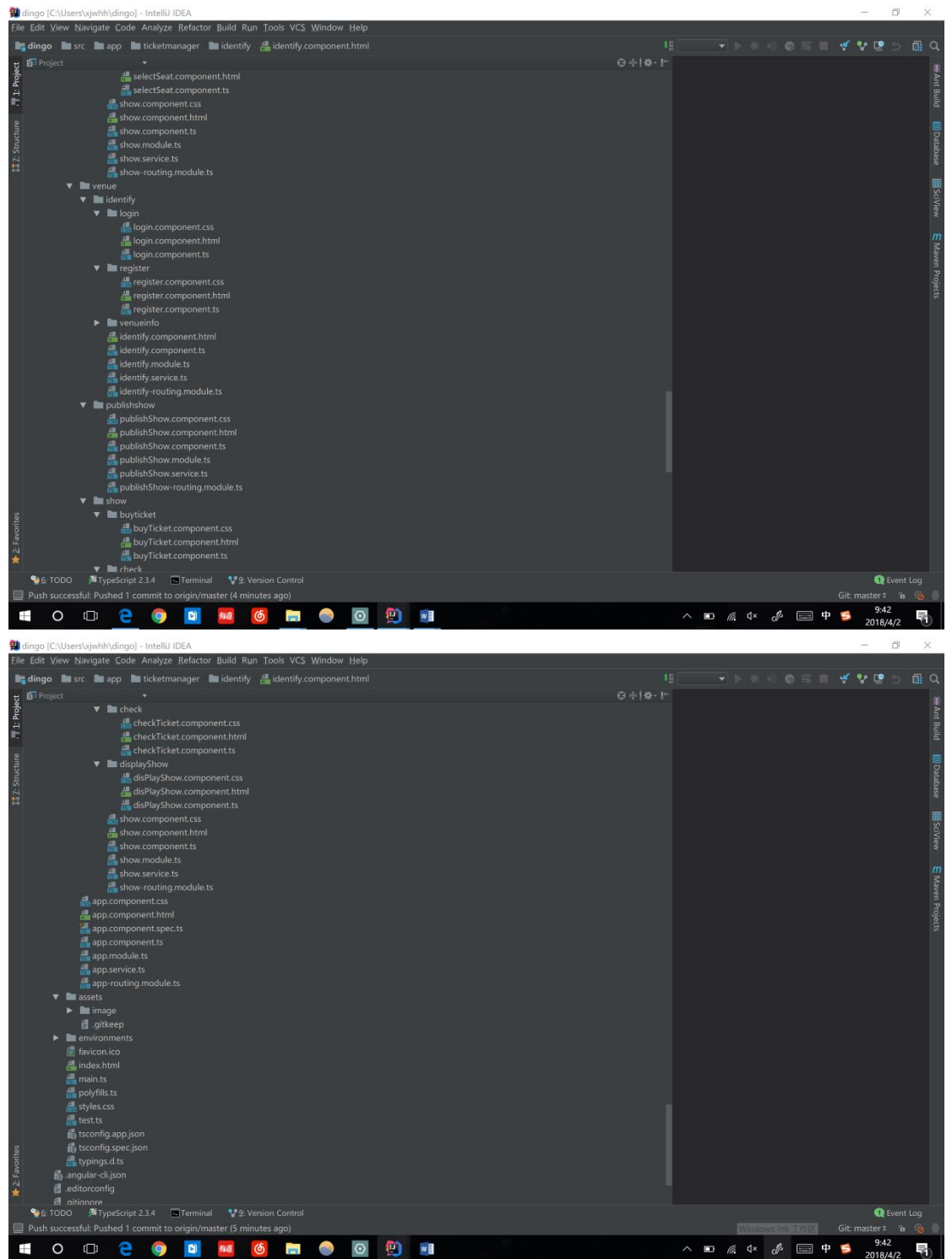
### 1. 工程的项目结构截图

#### 1) 前端



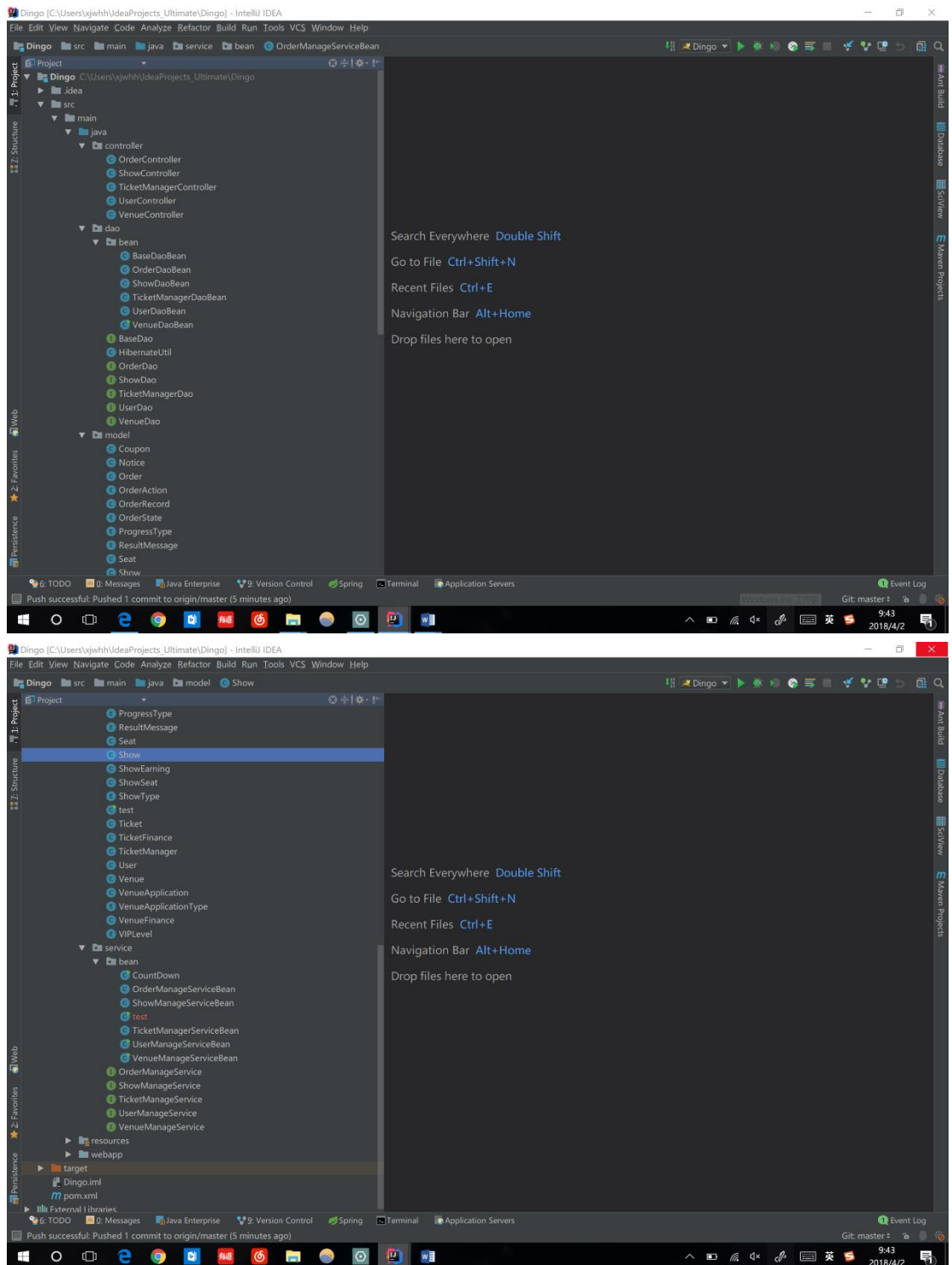






2) 后端





## 2. 后端框架

SpringMVC+Hibernate+Maven, MVC 的 view 部分由前端框架代替

## 3. 前端页面框架

Angular4+Bootstrap

## 三. 类设计

## 1. 后端各包的类

### 1) model 包：实体类

Coupon (优惠券)  
Notice (通知)  
Order (订单)  
OrderAction (订单行为)  
OrderRecord (订单记录)  
OrderState (订单状态)  
ProgressType (演出进度)  
ResultMessage (结果)  
Seat (场馆座位)  
ShowEarning (演出收益)  
ShowSeat (演出座位)  
ShowType (演出类型)  
Ticket (演出票)  
TicketFinance (网站财务)  
TicketManager (网站管理员)  
User (用户)  
Venue (场馆)  
VenueApplication (场馆申请)  
VenueApplcaytionType (场馆申请种类)  
VenueFinance (场馆财务)  
VIPLevel (VIP 等级)

### 2) controller 包：url 接口，调用对应 service

OrderController: 订单相关的接口，包括添加，付款，取消，获取，检票，线下购买  
ShowController: 演出相关的接口，包括发布，获取  
TicketManagerController: 网站管理员相关的接口，包括登录，审批场馆申请，结算收益，获取网站财务信息  
UserController: 用户相关的接口，包括注册，登录，邮箱验证，注销，获取，更新信息  
VenueController: 场馆相关的接口，包括注册，登录，更新，申请，获取财务信息

### 3) dao 包：数据库操作，被 service 调用

HibernateUtil: 用于获取 session，进行数据库操作  
BaseDao: 实现了一些通用的保存，更新，查找操作  
OrderDao, OrderDaoBean: 订单相关的数据库操作接口及其实现  
ShowDao, ShowDaoBean: 演出相关的数据库操作接口及其实现  
TciektManagerDao, TicketManagerDaoBean: 网站及管理员相关的数据库操作接口及其实现  
UserDao, UserDaoBean: 用户相关的数据库操作接口及其实现  
VeneuDao, VenueDaoBean: 场馆相关的数据库操作接口及其实现

- 4) service 包：服务，被 controller 调用，调用 dao
- OrderManageService,OrderManageServiceBean:订单相关的服务
  - ShowManageService,ShowManageServiceBean:演出相关的服务
  - TicketManageService,TicketManageServiceBean:网站管理员相关的服务
  - UserManageService,UserManageServiceBean:用户相关的服务
  - VenueManageService,VenueManageServiceBean:场馆相关的服务

## 2. 前端各页面

### 1) user:用户所见的页面

- identify:用户信息总界面
  - userInfo:用户信息界面
    - basicInfo:基本信息界面
    - orderInfo:订单信息界面
    - statisticsInfo:统计信息界面
    - updateInfo:更新基本信息界面
- register:注册界面
- login:登录界面
- emailConfirmation:邮箱认证界面
- pay:付款界面
- show:演出信息界面
  - displayShow:演出信息展示界面
  - noSelectSeat:不选座购买界面
  - selectSeat:选座购买界面

### 2) venue:场馆管理人员所见的页面

- identify:场馆信息总界面
  - userInfo:场馆信息界面
    - basicInfo:基本信息界面
    - orderInfo:订单信息界面
    - statisticsInfo:统计信息界面
    - updateInfo:更新基本信息界面
- register:注册界面
- login:登录界面
- publishShow:发布演出界面
- show:演出信息界面
  - displayShow:演出信息展示界面
  - checkTicket:检票界面
  - buyTicket:现场购票界面

### 3) ticketManager:网站管理人员所见的界面

- identify:网站信息总界面
  - login:登录界面
  - userInfo:网站管理员信息界面

basicInfo:网站管理员基本信息界面  
settle:结算收益界面  
examine:审批场馆申请  
update:审批更新信息申请  
register:审批场馆注册申请  
statisticsInfo:统计信息  
finance:财务统计界面  
venue:场馆统计界面  
user:用户统计界面

#### 四 . 其他

##### 1. 开发环境

数据库：mysql

服务器：tomcat

##### 2. 开发心得体会

熟悉了 SpringMVC 和 hibernate 框架，复习了 Angular4 框架

hibernate 相比 JDBC 更加方便，开发速度快

获得了前后端分离的项目经验

性能很重要