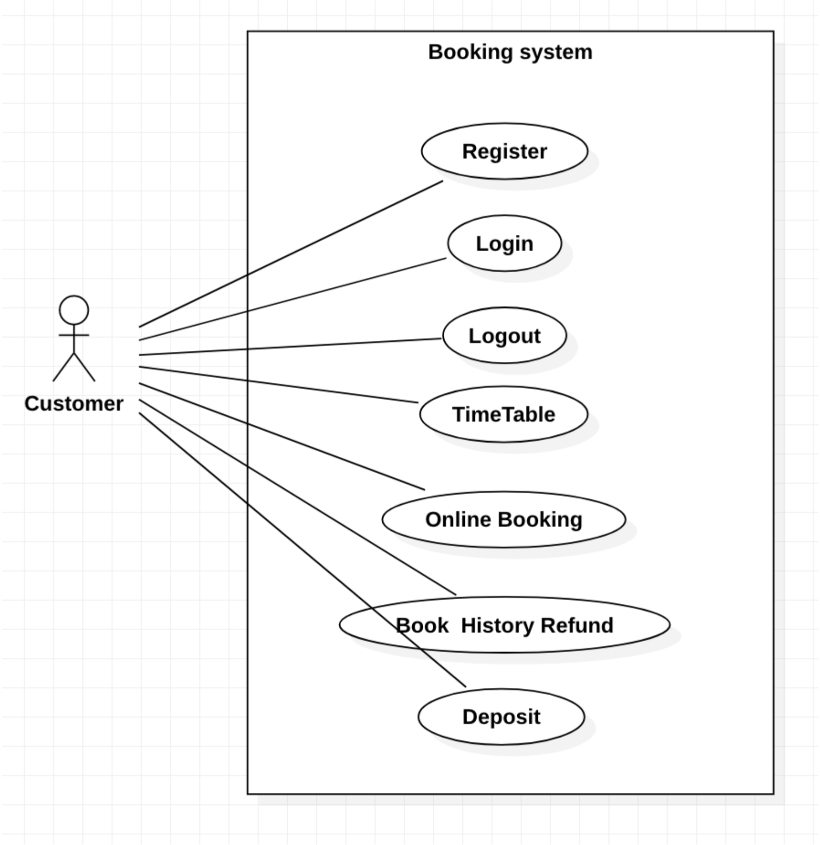
OOAD Final Project - Team report

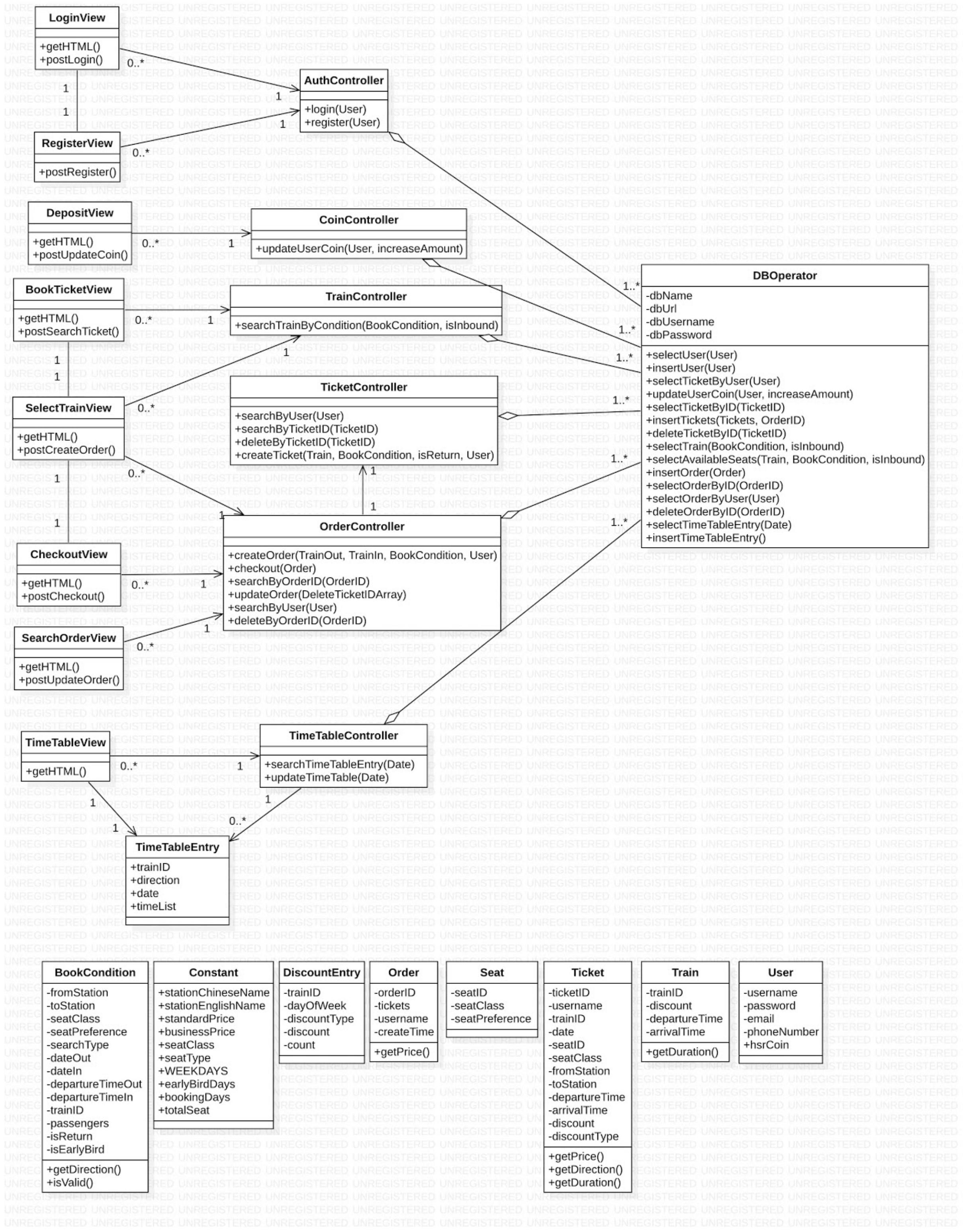
第一組 (報告順序3)

analysis&design

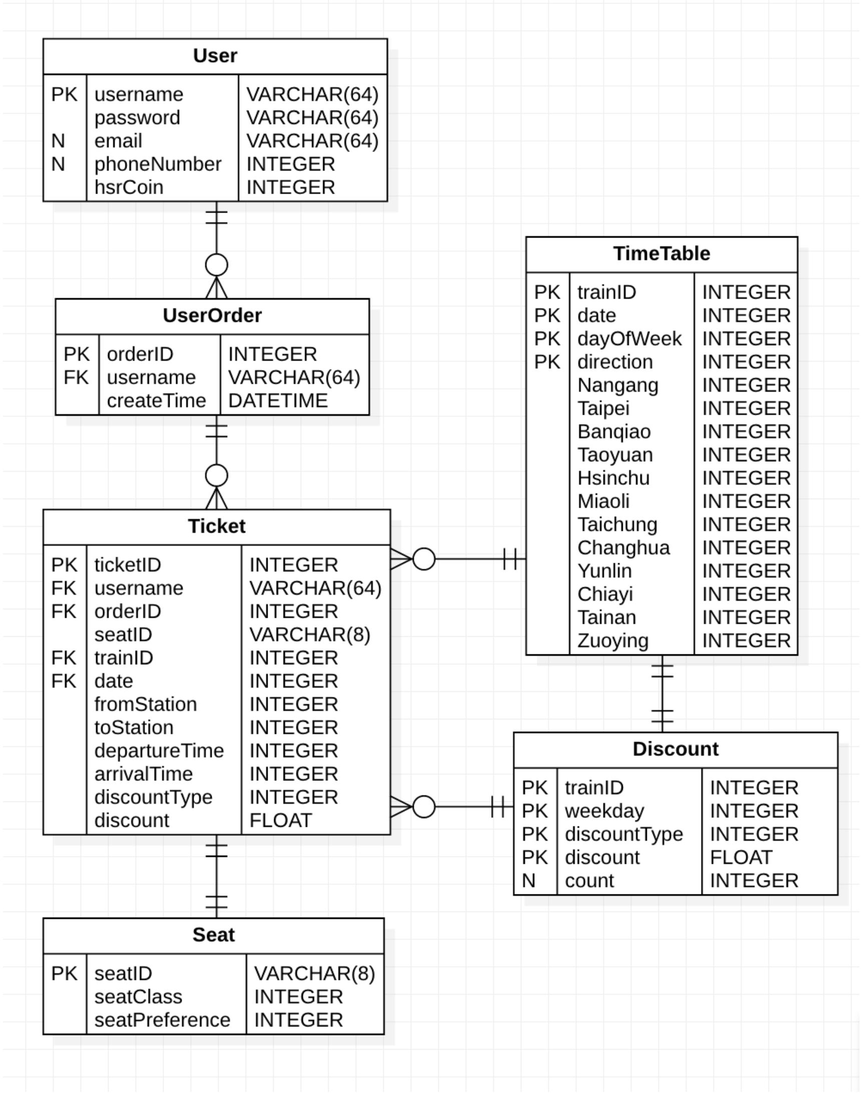
- Use Case



- Class Diagram



- Database (ER model)



User

PK	username	VARCHAR(64)
	password	VARCHAR(64)
N	email	VARCHAR(64)
N	phoneNumber	INTEGER
	hsrCoin	INTEGER

UserOrder

PK	orderID	INTEGER
FK	username	VARCHAR(64)
	createTime	DATETIME

Ticket

PK	ticketID	INTEGER
FK	username	VARCHAR(64)
FK	orderID	INTEGER
	seatID	VARCHAR(8)
FK	trainID	INTEGER
FK	date	INTEGER
	fromStation	INTEGER
	toStation	INTEGER
	departureTime	INTEGER
	arrivalTime	INTEGER
	discountType	INTEGER
	discount	FLOAT

Seat

PK	seatID	VARCHAR(8)
	seatClass	INTEGER
	seatPreference	INTEGER

TimeTable

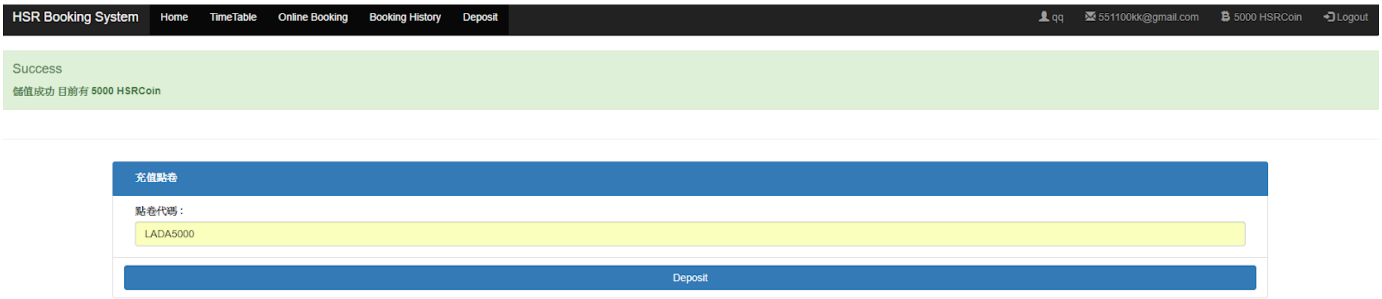
PK	trainID	INTEGER
PK	date	INTEGER
PK	dayOfWeek	INTEGER
PK	direction	INTEGER
	Nangang	INTEGER
	Taipei	INTEGER
	Banqiao	INTEGER
	Taoyuan	INTEGER
	Hsinchu	INTEGER
	Miaoli	INTEGER
	Taichung	INTEGER
	Changhua	INTEGER
	Yunlin	INTEGER
	Chiayi	INTEGER
	Tainan	INTEGER
	Zuoying	INTEGER

Discount

PK	trainID	INTEGER
PK	weekday	INTEGER
PK	discountType	INTEGER
PK	discount	FLOAT
N	count	INTEGER

- UI-Bounding

儲值



Success

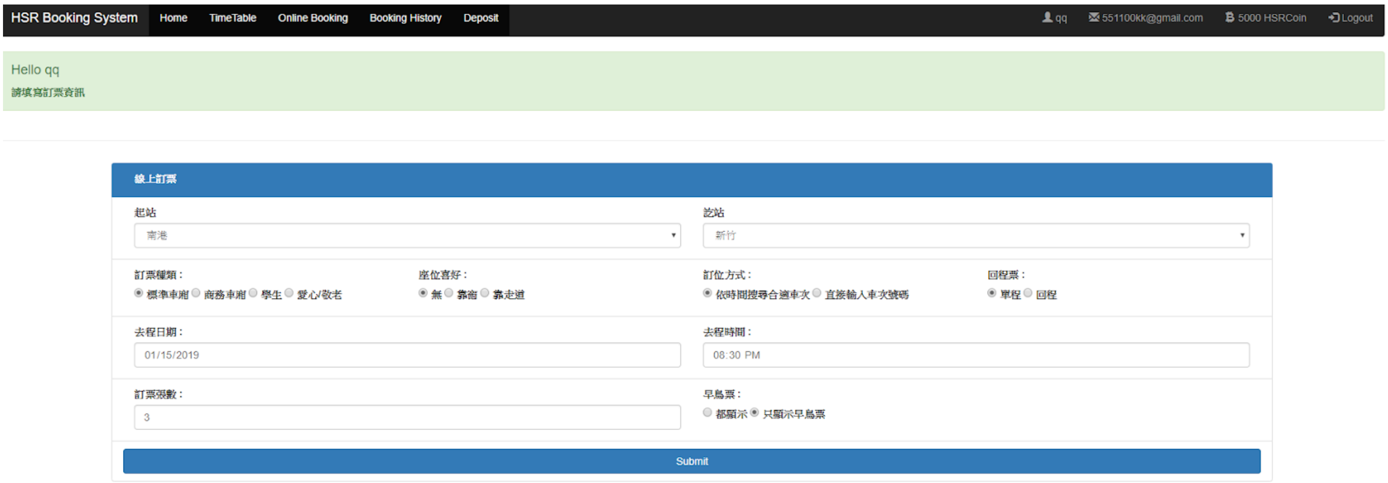
儲值成功 目前有 5000 HSRCoin

充值點卷

點卷代碼：

LADA5000

訂票



Hello qq

請填寫訂票資訊

線上訂票

起站

南港

訖站

新竹

訂票種類：◉ 標準車廂 ○ 商務車廂 ○ 學生 ○ 愛心/敬老

座位喜好：◉ 無 ○ 靠窗 ○ 靠走道

訂位方式：◉ 依時間搜尋合適車次 ○ 直接輸入車次號碼

回程票：◉ 單程 ○ 回程

去程日期：

01/15/2019

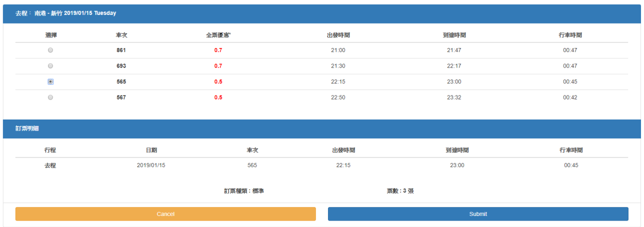
去程時間：

08:30 PM

訂票張數：

3

早鳥票：○ 都顯示 ◉ 只顯示早鳥票



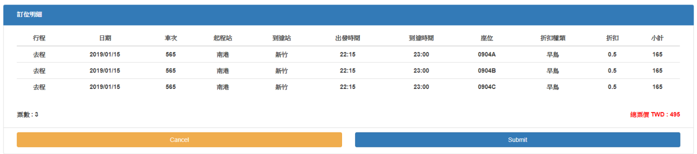
去程： 南港 - 新竹 2019/01/15 Tuesday

選擇	車次	全票優惠*	出發時間	到達時間	行車時間
○	861	0.7	21:00	21:47	00:47
○	693	0.7	21:30	22:17	00:47
◉	565	0.5	22:15	23:00	00:45
○	567	0.5	22:50	23:32	00:42

訂票明細

行程	日期	車次	出發時間	到達時間	行車時間
去程	2019/01/15	565	22:15	23:00	00:45

訂票種類：標準 票數：3 張

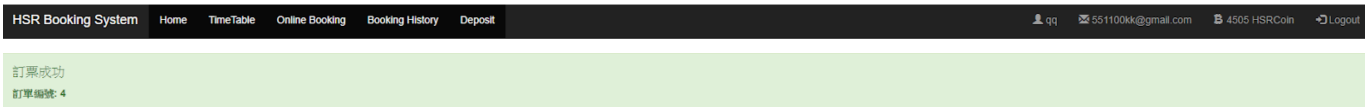


訂位明細

行程	日期	車次	起程站	到達站	出發時間	到達時間	座位	折扣種類	折扣	小計
去程	2019/01/15	565	南港	新竹	22:15	23:00	0904A	早鳥	0.5	165
去程	2019/01/15	565	南港	新竹	22:15	23:00	0904B	早鳥	0.5	165
去程	2019/01/15	565	南港	新竹	22:15	23:00	0904C	早鳥	0.5	165

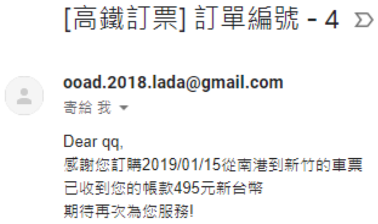
票數：3

總票價 TWD：495



訂票成功

訂單編號: 4



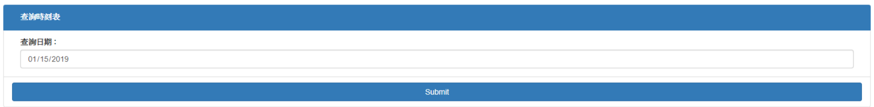
[高鐵訂票] 訂單編號 - 4

ooad.2018.lada@gmail.com

寄給 我

Dear qq,
感謝您訂購2019/01/15從南港到新竹的車票
已收到您的帳款495元新台幣
期待再次為您服務!

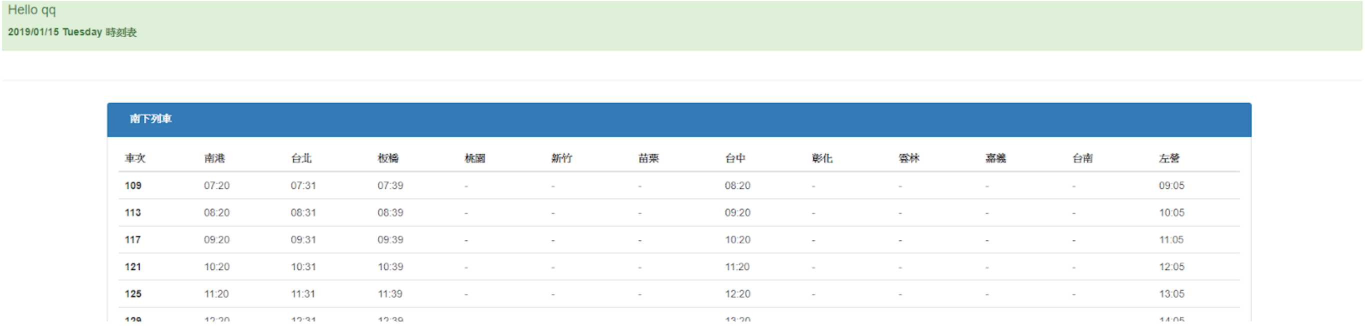
時刻表



查詢時刻表

查詢日期：

01/15/2019



Hello qq

2019/01/15 Tuesday 時刻表

南下列車

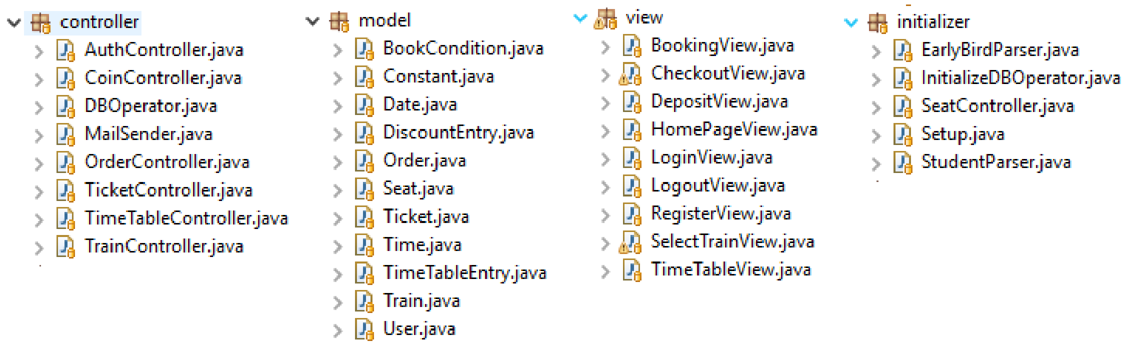
車次	南港	台北	板橋	桃園	新竹	苗栗	台中	彰化	雲林	嘉義	台南	左營
109	07:20	07:31	07:39	-	-	-	08:20	-	-	-	-	09:05
113	08:20	08:31	08:39	-	-	-	09:20	-	-	-	-	10:05
117	09:20	09:31	09:39	-	-	-	10:20	-	-	-	-	11:05
121	10:20	10:31	10:39	-	-	-	11:20	-	-	-	-	12:05
125	11:20	11:31	11:39	-	-	-	12:20	-	-	-	-	13:05
129	12:20	12:31	12:39				13:20					14:05

Implementation

套件:

- **json.jar:** 用來parse university discount和earlybird discount
- **jsoup.jar:** 用來parse高鐵官方網站的時刻表
- **mysql-connector, jdbc:** 用來連接並且操作mysql
- **mail.jar:** 當訂票成功時，寄信給使用者
- **rapid core.jar:** JSP的template架構
- **junit:** 用來做Unit test

MVC framework:



controller
- AuthController.java
- CoinController.java
- DBOperator.java
- MailSender.java
- OrderController.java
- TicketController.java
- TimeTableController.java
- TrainController.java

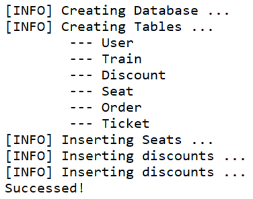
model
- BookCondition.java
- Constant.java
- Date.java
- DiscountEntry.java
- Order.java
- Seat.java
- Ticket.java
- Time.java
- TimeTableEntry.java
- Train.java
- User.java

view
- BookingView.java
- CheckoutView.java
- DepositView.java
- HomePageView.java
- LoginView.java
- LogoutView.java
- RegisterView.java
- SelectTrainView.java
- TimeTableView.java

initializer
- EarlyBirdParser.java
- InitializeDBOperator.java
- SeatController.java
- Setup.java
- StudentParser.java

Setup:

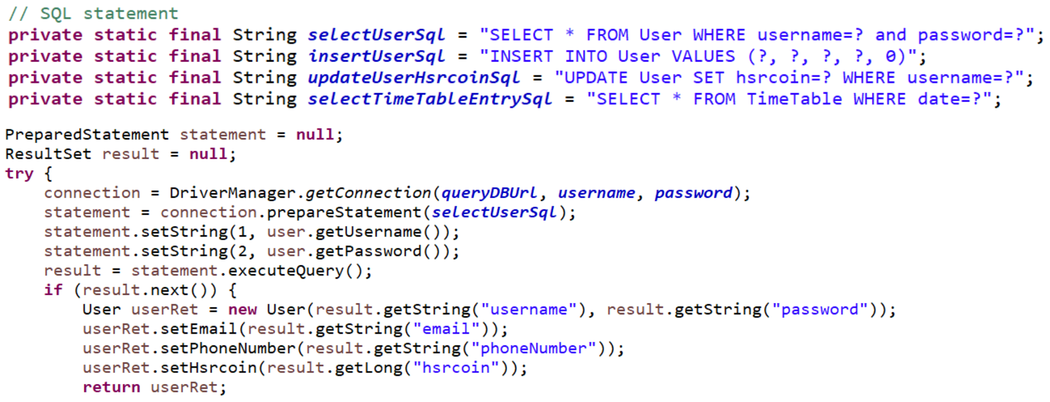
對於第一次執行Server的機器，必須先執行Setup來進行create DB table



```
[INFO] Creating Database ...
[INFO] Creating Tables ...
        --- User
        --- Train
        --- Discount
        --- Seat
        --- Order
        --- Ticket
[INFO] Inserting Seats ...
[INFO] Inserting discounts ...
[INFO] Inserting discounts ...
Successed!
```

Controller/DBOerator.java:

所有和mysql有關的語句都存放在這個Class裡，所有的Statement都**使用PreparedStatement**，**不使用String.format**，避免SqlInjection的漏洞。



```
// SQL statement
private static final String selectUserSql = "SELECT * FROM User WHERE username=? and password=?";
private static final String insertUserSql = "INSERT INTO User VALUES (?, ?, ?, ?, 0)";
private static final String updateUserHsrcoinSql = "UPDATE User SET hsrcoin=? WHERE username=?";
private static final String selectTimeTableEntrySql = "SELECT * FROM TimeTable WHERE date=?";

PreparedStatement statement = null;
ResultSet result = null;
try {
    connection = DriverManager.getConnection(queryDBUrl, username, password);
    statement = connection.prepareStatement(selectUserSql);
    statement.setString(1, user.getUsername());
    statement.setString(2, user.getPassword());
    result = statement.executeQuery();
    if (result.next()) {
        User userRet = new User(result.getString("username"), result.getString("password"));
        userRet.setEmail(result.getString("email"));
        userRet.setPhoneNumber(result.getString("phoneNumber"));
        userRet.setHsrcoin(result.getLong("hsrcoin"));
        return userRet;
```

Multiple clients:

很多使用者可以同時連到WebServer，Tomcat Servlet在處理Request的時候是Asynchronized，只有在Critical Section的部分要使用mutex。

Unit Test:

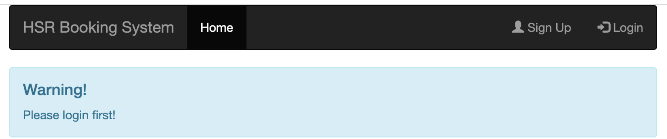
針對特定功能的function，輸入input用assert確定返回正確的return值

Cache:

把TimeTable cache在local，使得再次查詢時可以更快。

Demo

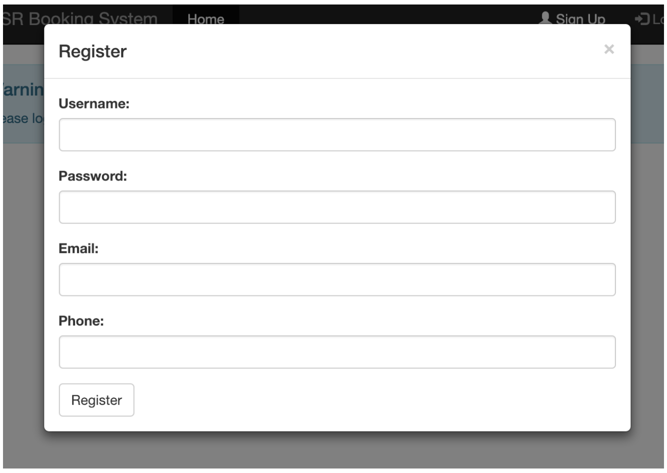
Home(未登入)



Warning!

Please login first!

Register(註冊)



Register

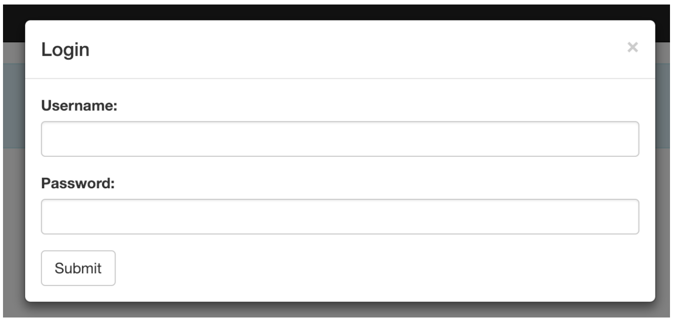
Username:

Password:

Email:

Phone:

Login(登入)

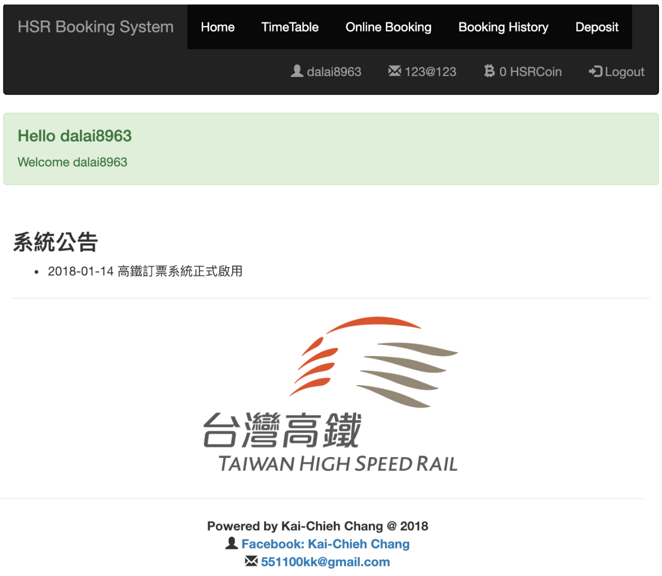


Login

Username:

Password:

Home(已登入)



Hello dalai8963

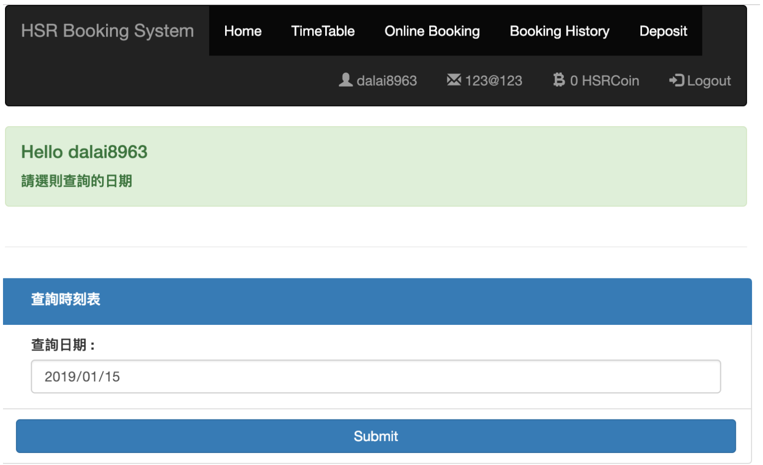
Welcome dalai8963

系統公告

- 2018-01-14 高鐵訂票系統正式啟用

Powered by Kai-Chieh Chang @ 2018
Facebook: Kai-Chieh Chang
551100kk@gmail.com

TimeTable(時刻表查詢)



Hello dalai8963

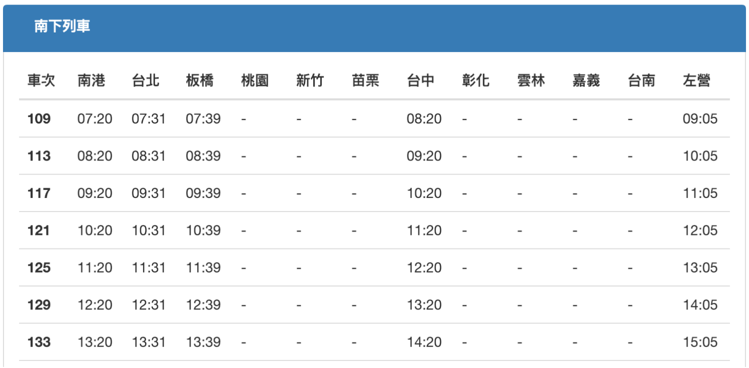
請選則查詢的日期

查詢時刻表

查詢日期：

2019/01/15

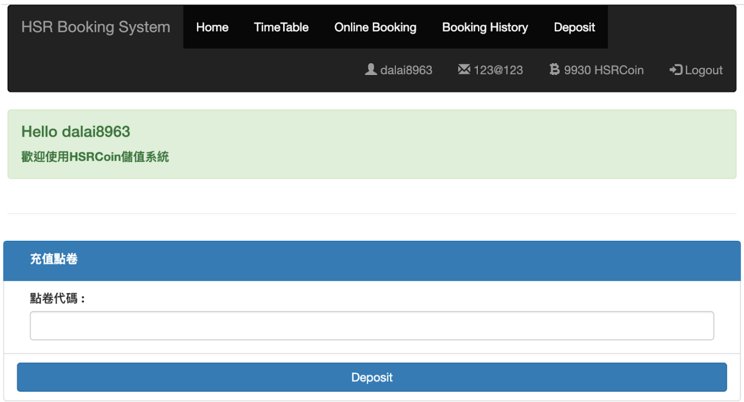
時刻表查詢結果



南下列車

車次	南港	台北	板橋	桃園	新竹	苗栗	台中	彰化	雲林	嘉義	台南	左營
109	07:20	07:31	07:39	-	-	-	08:20	-	-	-	-	09:05
113	08:20	08:31	08:39	-	-	-	09:20	-	-	-	-	10:05
117	09:20	09:31	09:39	-	-	-	10:20	-	-	-	-	11:05
121	10:20	10:31	10:39	-	-	-	11:20	-	-	-	-	12:05
125	11:20	11:31	11:39	-	-	-	12:20	-	-	-	-	13:05
129	12:20	12:31	12:39	-	-	-	13:20	-	-	-	-	14:05
133	13:20	13:31	13:39	-	-	-	14:20	-	-	-	-	15:05

Deposit(儲值)



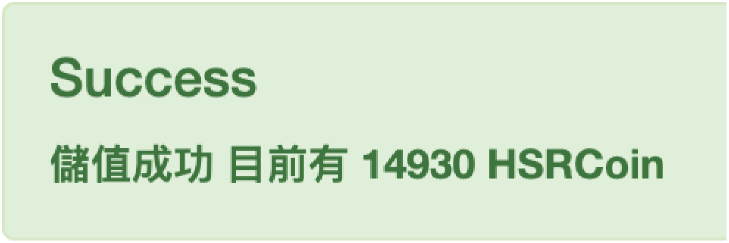
Hello dalai8963

歡迎使用HSRCoin儲值系統

充值點卷

點卷代碼：

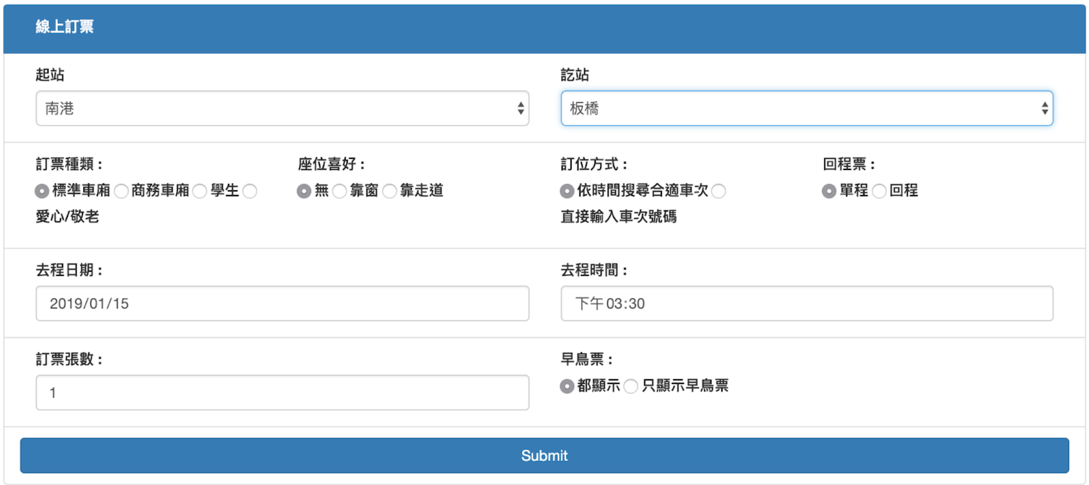
儲值成功



Success

儲值成功 目前有 14930 HSRCoin

Online Booking(線上訂票)



線上訂票

起站
南港

訖站
板橋

訂票種類：
● 標準車廂 ○ 商務車廂 ○ 學生 ○ 愛心/敬老

座位喜好：
● 無 ○ 靠窗 ○ 靠走道

訂位方式：
● 依時間搜尋合適車次 ○ 直接輸入車次號碼

回程票：
● 單程 ○ 回程

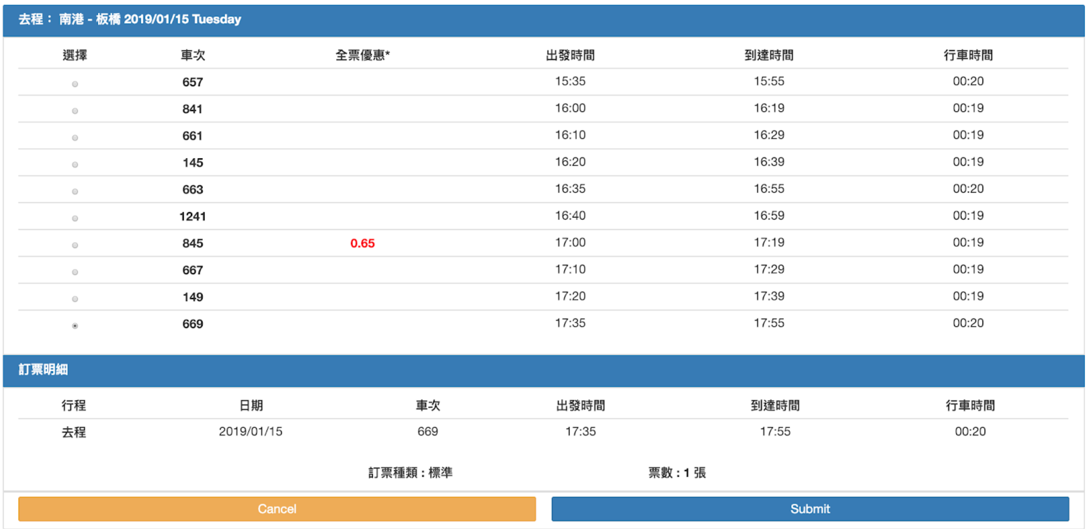
去程日期：
2019/01/15

去程時間：
下午03:30

訂票張數：
1

早鳥票：
● 都顯示 ○ 只顯示早鳥票

選擇車次與訂單明細



去程： 南港 - 板橋 2019/01/15 Tuesday

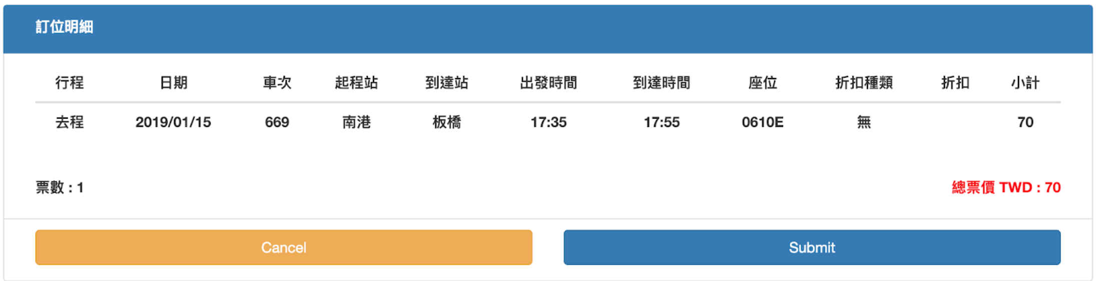
選擇	車次	全票優惠*	出發時間	到達時間	行車時間
○	657		15:35	15:55	00:20
○	841		16:00	16:19	00:19
○	661		16:10	16:29	00:19
○	145		16:20	16:39	00:19
○	663		16:35	16:55	00:20
○	1241		16:40	16:59	00:19
○	845	0.65	17:00	17:19	00:19
○	667		17:10	17:29	00:19
○	149		17:20	17:39	00:19
●	669		17:35	17:55	00:20

訂票明細

行程	日期	車次	出發時間	到達時間	行車時間
去程	2019/01/15	669	17:35	17:55	00:20

訂票種類：標準 票數：1 張

訂位明細



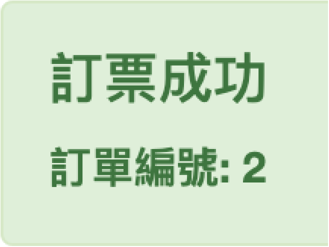
訂位明細

行程	日期	車次	起程站	到達站	出發時間	到達時間	座位	折扣種類	折扣	小計
去程	2019/01/15	669	南港	板橋	17:35	17:55	0610E	無		70

票數：1

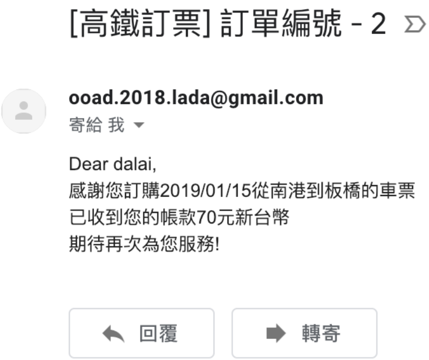
總票價 TWD：70

訂票成功資訊＆信箱收到訂票成功資訊



訂票成功

訂單編號: 2

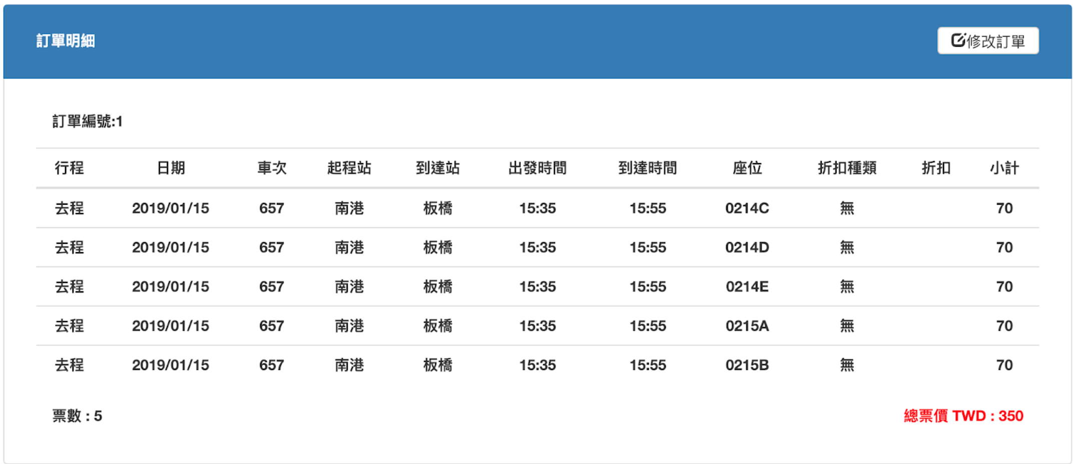


[高鐵訂票] 訂單編號 - 2

ooad.2018.lada@gmail.com
寄給 我

Dear dalai,
感謝您訂購2019/01/15從南港到板橋的車票
已收到您的帳款70元新台幣
期待再次為您服務!

Booking History(訂票記錄)



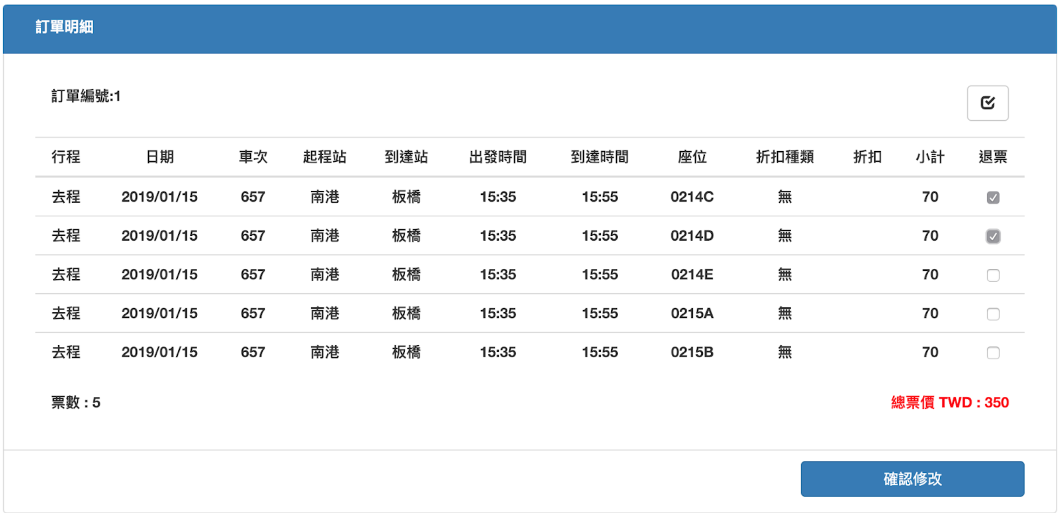
訂單編號:1

行程	日期	車次	起程站	到達站	出發時間	到達時間	座位	折扣種類	折扣	小計
去程	2019/01/15	657	南港	板橋	15:35	15:55	0214C	無		70
去程	2019/01/15	657	南港	板橋	15:35	15:55	0214D	無		70
去程	2019/01/15	657	南港	板橋	15:35	15:55	0214E	無		70
去程	2019/01/15	657	南港	板橋	15:35	15:55	0215A	無		70
去程	2019/01/15	657	南港	板橋	15:35	15:55	0215B	無		70

票數 : 5

總票價 TWD : 350

修改訂單(前)



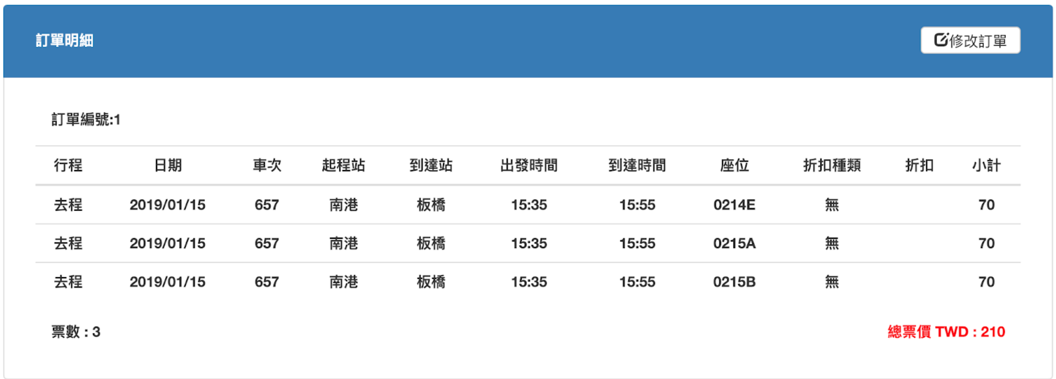
訂單編號:1

行程	日期	車次	起程站	到達站	出發時間	到達時間	座位	折扣種類	折扣	小計	退票
去程	2019/01/15	657	南港	板橋	15:35	15:55	0214C	無		70	☑
去程	2019/01/15	657	南港	板橋	15:35	15:55	0214D	無		70	☑
去程	2019/01/15	657	南港	板橋	15:35	15:55	0214E	無		70	□
去程	2019/01/15	657	南港	板橋	15:35	15:55	0215A	無		70	□
去程	2019/01/15	657	南港	板橋	15:35	15:55	0215B	無		70	□

票數 : 5

總票價 TWD : 350

修改訂單(後)



訂單編號:1

行程	日期	車次	起程站	到達站	出發時間	到達時間	座位	折扣種類	折扣	小計
去程	2019/01/15	657	南港	板橋	15:35	15:55	0214E	無		70
去程	2019/01/15	657	南港	板橋	15:35	15:55	0215A	無		70
去程	2019/01/15	657	南港	板橋	15:35	15:55	0215B	無		70

票數 : 3

總票價 TWD : 210

Comparison

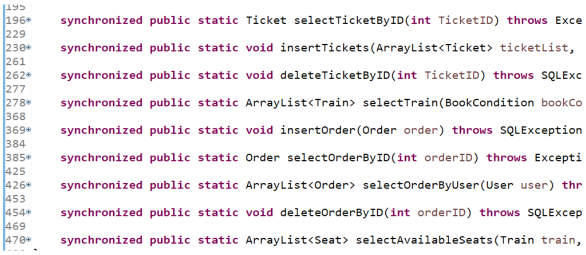
	Sample	HSR-Booking©
UI	又醜又髒的swing	Bootstrap
Server	無	Client-Server
Multi-client	無	允許多Client
登入系統	無	有
儲值/支付系統	無	HSR Coin
訂單成功寄信	無	貼心提醒
MVC Framework	無	大勝
DAO	無	安全
跨平台	PC only	全面支援
客戶端	必須安裝JRE	多數瀏覽器即可

Reuse

- **TimeTableParser**
    - 收集時間表的功能可以重覆使用在其他的應用

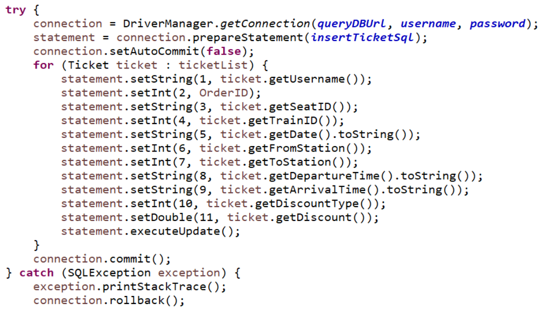
Pattern

- **Mutex & Read, Write Lock**
    - 在寫入資料庫時的Code是一段Critical section，若同時進入則有可能造成不可預期的錯誤，例如： 同時搶一個座位、同時更新時刻表



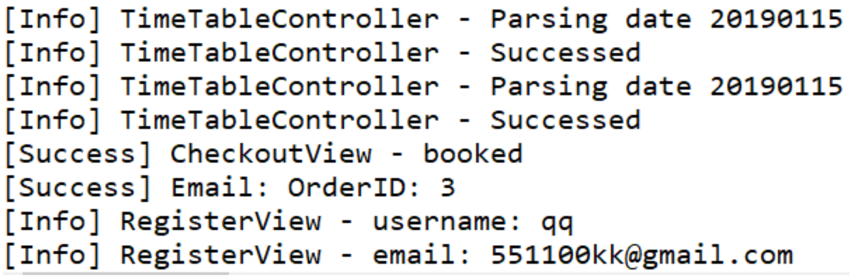
```
    synchronized public static Ticket selectTicketByID(int TicketID) throws Exce

    synchronized public static void insertTickets(ArrayList<Ticket> ticketList,

    synchronized public static void deleteTicketByID(int TicketID) throws SQLExc

    synchronized public static ArrayList<Train> selectTrain(BookCondition bookCo

    synchronized public static void insertOrder(Order order) throws SQLException

    synchronized public static Order selectOrderByID(int orderID) throws Excepti

    synchronized public static ArrayList<Order> selectOrderByUser(User user) thr

    synchronized public static void deleteOrderByID(int orderID) throws SQLExcep

    synchronized public static ArrayList<Seat> selectAvailableSeats(Train train,
```

- **Transaction:** Insert Order的ticket array的時候，若其中一張失敗則必須roll back。



```
try {
    connection = DriverManager.getConnection(queryDBUrl, username, password);
    statement = connection.prepareStatement(insertTicketSql);
    connection.setAutoCommit(false);
    for (Ticket ticket : ticketList) {
        statement.setString(1, ticket.getUsername());
        statement.setInt(2, OrderID);
        statement.setString(3, ticket.getSeatID());
        statement.setInt(4, ticket.getTrainID());
        statement.setString(5, ticket.getDate().toString());
        statement.setInt(6, ticket.getFromStation());
        statement.setInt(7, ticket.getToStation());
        statement.setString(8, ticket.getDepartureTime().toString());
        statement.setString(9, ticket.getArrivalTime().toString());
        statement.setInt(10, ticket.getDiscountType());
        statement.setDouble(11, ticket.getDiscount());
        statement.executeUpdate();
    }
    connection.commit();
} catch (SQLException exception) {
    exception.printStackTrace();
    connection.rollback();
```

- **Database Normalization**
    - 資料庫Table的資料不存放可以select出來的資料，善用Join table來獲得資料，例如： 要查詢某一班車次在某一段還有哪些空位，則可以select出Seat與User，並且取叉集。
    - selectAvailableSeatSql = "SELECT * FROM Seat WHERE seatID not in (SELECT seatid FROM Ticket where trainID=? and date=?) AND seatClass=?
- **Unit test - result test**
    - 用unit test來測試各個unit正常運作
- **Cache-Aside**
    - 把Timetable cache 在local，第二次以後查詢會快很多
- **Log**
    - 透過紀錄Log可以在發生錯誤時方便Debug
    -



```
[Info] TimeTableController - Parsing date 20190115
[Info] TimeTableController - Successed
[Info] TimeTableController - Parsing date 20190115
[Info] TimeTableController - Successed
[Success] CheckoutView - booked
[Success] Email: OrderID: 3
[Info] RegisterView - username: qq
[Info] RegisterView - email: 551100kk@gmail.com
```

Coupling coherent & Modularization & Componentization

- 模組化依照MVC架構
  View細分成Login,Deposit,Bookticket…等等
  Model依照實體的物件來建構，例如ticket,user,order…
  Controller對應各種View與Model和Datebase來做操作