

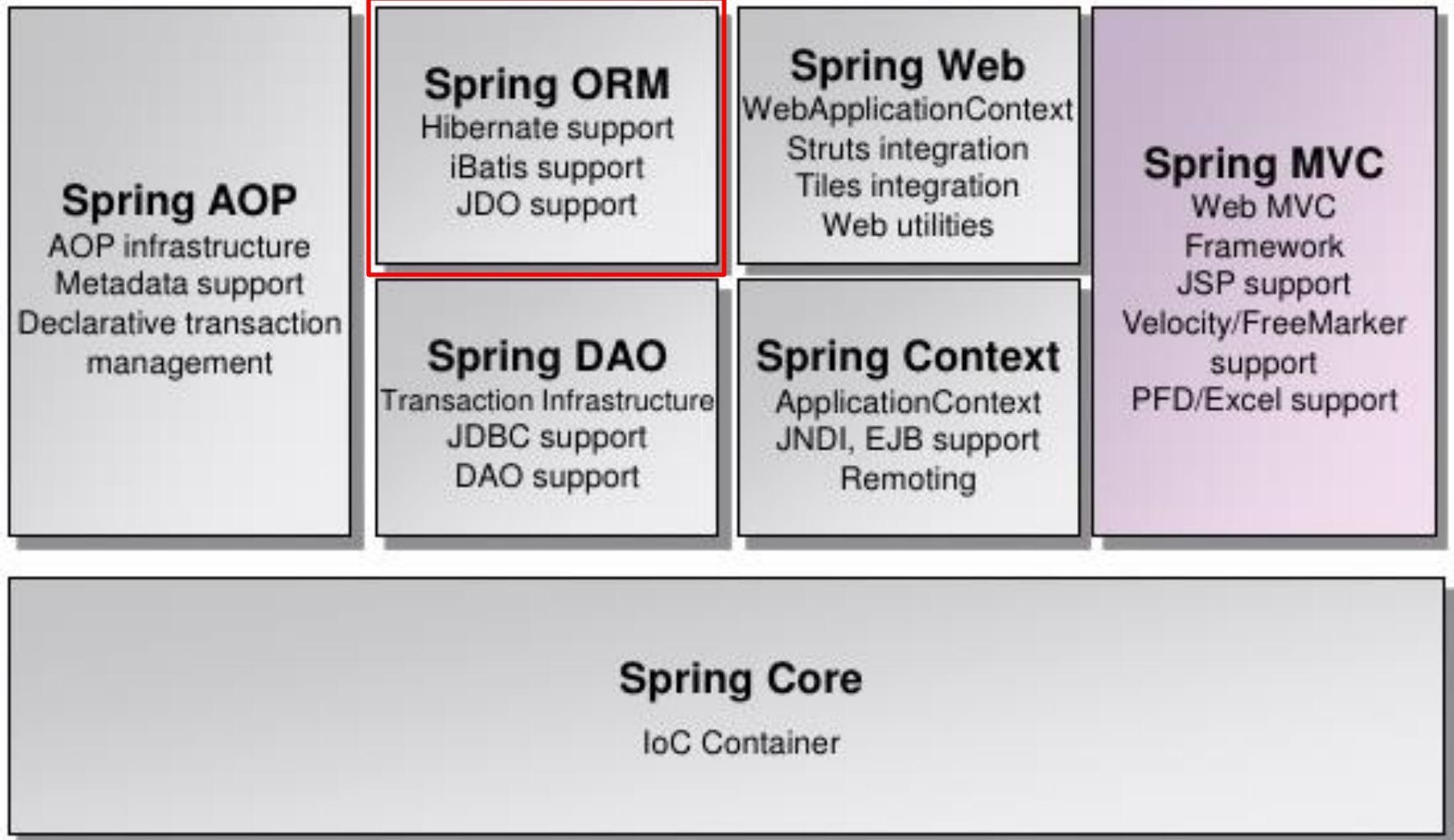


# TÍCH HỢP HIBERNATE

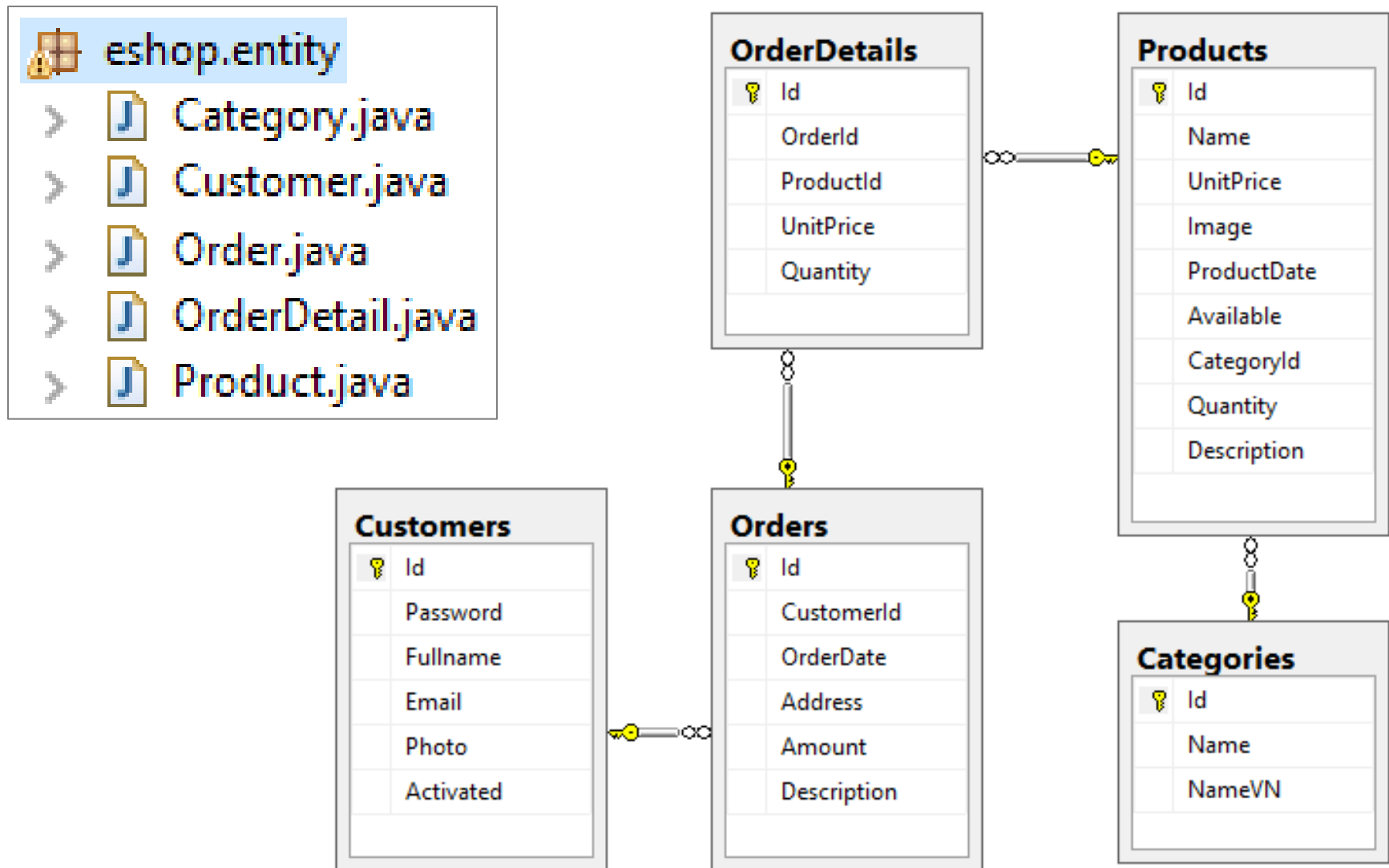
ThS. Nguyễn Nghiê**̣**m  
0913.745.789 - NghiemN@fpt.edu.vn

- Giới thiệu
- Tích hợp Hibernate
- Cơ chế điều khiển transaction
- Hoàn thiện AccountController

# GIỚI THIỆU



# REVIEW ESHOPV10



*Xem lại phần bài học Hibernate để hiểu rõ việc ánh xạ thực thể*



# REVIEW HIBERNATE

---

- Khi làm việc với Hibernate chúng ta cần
  - ✱ Hibernate.cfg.xml
  - ✱ Entity class
- Lập trình với Hibernate
  - ✱ Nạp file cấu hình
  - ✱ Tạo SessionFactory
  - ✱ Tạo Session
  - ✱ Truy vấn và thao tác



# TÍCH HỢP HIBERNATE

---

- Khai báo thông số kết nối, SessionFactory và Transaction trong file cấu hình
- Tiêm SessionFactory vào Controller để sử dụng
- Lập trình truy vấn và thao tác như cũ
- Điều khiển transaction chung hoặc riêng tùy vào tình huống
- Sử dụng chung một session sẽ tăng tính performance của ứng dụng

# KHAI BÁO CẤU HÌNH HIBERNATE

<!-- DataSource -->

```
<bean id="dataSource"
      class="org.springframework.jdbc.datasource.DriverManagerDataSource"
      p:driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
      p:url="jdbc:sqlserver://localhost; Database=EshopV10"
      p:username="sa" p:password="songlong">
</bean>
```

Thông số kết nối CSDL

<!-- Hibernate Session Factory -->

```
<bean id="sessionFactory"
      class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
  <property name="dataSource" ref="dataSource"/>
  <property name="hibernateProperties">
    <props>
      <prop key="hibernate.dialect">org.hibernate.dialect.SQLServerDialect</prop>
      <prop key="hibernate.show_sql">>false</prop>
    </props>
  </property>
  <property name="packagesToScan" value="eshop.entity"/>
</bean>
```

Package chứa entity class

<!-- Transaction Manager -->

```
<tx:annotation-driven transaction-manager="transactionManager" />
<bean id="transactionManager"
      class="org.springframework.orm.hibernate4.HibernateTransactionManager"
      p:sessionFactory-ref="sessionFactory" />
```

# TỰ ĐIỀU KHIỂN TRANSACTION

```
@Controller
@RequestMapping("account")
public class AccountController {
    @Autowired
    SessionFactory factory;
```

Tiêm SessionFactory

```
@RequestMapping(value="register", method=RequestMethod.POST)
public String register(ModelMap model,
    @ModelAttribute("user") Customer user) {
    Session session = factory.openSession();
    Transaction transaction = session.beginTransaction();
    try {
        session.save(user);
        transaction.commit();
        model.addAttribute("message", "Đăng ký thành công !");
    }
    catch (Exception e) {
        transaction.rollback();
        model.addAttribute("message", "Đăng ký thất bại !");
        e.printStackTrace();
    }
    session.close();
    return "register";
}
```



# TRANSACTION TỰ ĐỘNG THEO ACTION

@Controller

@RequestMapping("account")

**public class** AccountController {

@Autowired

SessionFactory factory;

Tự động điều khiển  
transaction cho action này

@Transactional

@RequestMapping(value="register", method=RequestMethod.**POST**)

**public** String register(ModelMap model,

    @ModelAttribute("user") Customer user) {

    Session session = factory.getCurrentSession();

    session.save(user);

**return** "register";

}

}

# TRANSACTION TỰ ĐỘNG TẤT CẢ ACTION

```
@Controller
@RequestMapping("account")
@Transactional
public class AccountController {
    @Autowired
    SessionFactory factory;
```

Tự động điều khiển transaction  
cho tất cả các action

```
@RequestMapping(value="register", method=RequestMethod.POST)
public String register(ModelMap model,
    @ModelAttribute("user") Customer user) {
    Session session = factory.getCurrentSession();
    session.save(user);
    return "register";
}
```

```
@RequestMapping(value="update", method=RequestMethod.POST)
public String update(ModelMap model,
    @ModelAttribute("user") Customer user) {
    Session session = factory.getCurrentSession();
    session.update(user);
    return "register";
}
```

```
}
```

## ● Hoàn thiện AccountController với các action sau

- ✱ Đăng nhập
- ✱ Đăng ký có upload hình và gửi email kích hoạt tài khoản
- ✱ Kích hoạt tài khoản sau đăng ký
- ✱ Quên mật khẩu gửi mật khẩu qua email
- ✱ Đăng xuất
- ✱ Đổi mật khẩu
- ✱ Sửa đổi thông tin tài khoản tài khoản

- Giới thiệu
- Tích hợp Hibernate
- Cơ chế điều khiển transaction
- Hoàn thiện AccountController

