



École Polytech Tours
64 Avenue Jean Portalis
37200, TOURS ,FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Département Informatique

Rapport de stage de fin d' études 2016-17
—stage à YUANSHEN Software Co. , Ltd en Chine

Entreprise: YUANSHEN Software Co. , Ltd, 255 rue Huaihai, 221000 Xuzhou, Chine

Étudiant (Auteur): LI Yuanyuan DI3, Année 2016 - 2017, yuanyuan.li@etu.univ-tours.fr

Tuteur entreprise: LI Chengzhi, Directeur, lcz@zenyuit.com

Tuteur académique: Mathieu Delalandre , mathieu.delalandre@univ-tours.fr

Sommaire

1. Introduction.....	1
1.1 L' introduction de YUANSHEN.....	1
1.2 L' introduction d' équipe << CRM >>.....	1
2. Entraînement (03.07.2017 - 29.07.2017)	2
2.1 L' entité - <<User>>.....	2
2.1.1 Les étapes.....	2
2.1.2 Les attributs.....	2
2.1.3 L' algorithme.....	2
2.1.4 Diagramme de séquence - <<User>>.....	3
2.1.5 Diagramme de classe - <<User>>.....	3
2.2 L' entité - <<Book>>.....	4
2.2.1 Explication de classe <<Book>>.....	4
2.2.2 La classe - <<Book>>.....	5
2.2.3 Diagramme de classe - <<Book>>.....	5
2.3 L' entité - <<Demande>>.....	6
2.3.1 Explication de la classe <<Demande>>.....	6
2.3.2 La classe - << DemandeServiceImpl >>.....	6
2.3.3 Diagramme de classe - << Demande >>.....	7
3. Travail (31.07.2017-25.08.2017).....	8
3.1 Programme - Livres en ligne.....	8
3.1.1 Modèle conceptuel de données.....	8
3.1.2 La strcture de programme.....	8
3.1.3 Le résultat.....	10
3.2 Programme — SAILIAN.....	12
3.2.1 Introduction de SAILIAN.....	12
3.2.2 Missions.....	13
3.2.3 Note.....	13
4. Conclusion.....	14
5. Référence	15

1. Introduction

1.1 L ' introduction de YUANSHEN

- Nom d ' entreprise: YUANSHEN Software Co. , Ltd.

- Horaire de couverture: du lundi au samedi de 8h 30 à 17h 30.

- Adresse: chambre 502, bâtiment Gongjiaoshangmao, 255 rue Huaihai ouest, à Xuzhou, en Chine.

- Service:

Il Principalement engagés dans le développement, le consultation, la vente de software. Dans la région de Xuzhou, Xuzhou Yanshen Software Co., Ltd. (Xuzhou YSSOFT CO., LTD) est la plus grande entreprise qui s ' occupe de développement des système de gestion de l ' information. Il sert le gouvernement, l ' organisme sans but lucratif et des entreprises. Maintenant, Il a plus de 4800 clients. Dans la région de Xuzhou, il prédomine dans le domaine de gestion financière, de gestion de fonds, de planification des ressources des organisations dans les petites et les moyennes entreprises.

- Notion de service: professionnel, à temps, sincère.

1.2 L ' introduction d ' équipe << CRM >>

- CRM: CRM correspond à << Customer Relationship Management >> en anglais, en français la gestion de la relation client (GRC) . GRC est l ' ensemble des outils et techniques destinés à capter, traiter, analyser les informations relatives aux clients et aux prospects. Il peut aider à la prise de décision pour des affaires financières, améliorer procédure d ' entreprise. Ce n ' est pas seulement un logiciel, c ' est une combinaison de méthodologie, de logiciels et de capacités informatiques.

- L ' équipe CRM: il y a deux équipes dans le département R&D (recherche et développement) d ' entreprise. Un sert le gouvernement et l ' organisme sans but lucratif. Autre sert des entreprise qui s ' appelle équipe CRM. L ' équipe CRM principalement développe système de gestion de la relation de client dans l ' entreprise.

- Le directeur de l ' équipe CRM: monsieur SHI.

- Employé de l ' équipe CRM: 11

2. Entraînement (03.07.2017 - 29.07.2017)

Mon tuteur met moi à l' équipe de CRM qui développe web et application de gestion de la relation de client, monsieur SHI donne moi un petit programme comme un entraînement.

- Nom de programme: Livres en ligne—un web pour vendre et acheter des livres.
- L' outil:
 - Eclipse Java EE IDE MARS 4.5.1 jdk1.7
 - Navicat: connecte Base de données
 - Tomcat: serveur.
- Le structure de programme: spring + spring MVC + mybatis.

2.1 L' entité - <<User>>

Au début, il y a juste une entité- <<User>>, une classe—<<User>> avec deux attributs:

userName	Le nom de client
password	code secret de client pour Login

Il y a aussi une fonction qui peut lister les informations de clients.

- Mes missions:

Compléter quatre fonctionnements:

- (1) Ajouter un client au tableau <<User>>.
- (2) Supprimer un client dans le tableau.
- (3) Mettre à jour les informations du client.
- (4) Lister les informations de clients dont les gestionnaires ont besoin.

2.1.1 Les étapes

- Compléter le document—test.xml: exécuter les actions de Base de données, par exemple: ajouter, supprimer, mettre à jour . . .
- Compléter les documents — <<UseDao>>, <<UserService>>, <<UserServiceImpl>>: l' appel de test.xml.
- Utiliser la classe <<test>> pour tester si le document—test.xml est correct.

2.1.2 Les attributs

J' ajoute quatre attributs:

userId (la première clé)	ID de client
entreprise	L' entreprise de client
createdate	Date de création
deleteDate	Date de annulation

2.1.3 L' algorithme

- Ajouter:

<insert id="insertUser">

```
Insert into user(USERID,USERNAME,PASSWORD,USERENTERPRISE,CREATEDATE,
DELETEDATE)
values ({userId,jdbcType=VARCHAR},{userName,jdbcType=VARCHAR},
#{passWord,jdbcType=VARCHAR},{userEnterprise,jdbcType=VARCHAR},{createDate,jdbc
Type=DATE},{deleteDate,jdbcType=DATE})
```

</insert>

- Supprimer:

```
<delete id="deleteUser">
```

```
delete from user where USERID=#{id}
```

</delete>

- Mettre à jour:

```
<update id="updateUser">
```

```
update user set USERNAME=#{userName,jdbcType=VARCHAR},
```

```
PASSWORD=#{passWord,jdbcType=VARCHAR},USERENTERPRISE=#{userEnterprise,jdbcType=
VARCHAR},CREATEDATE=#{createDate,jdbcType=DATE},
```

```
DELETEDATE=#{deleteDate,jdbcType=DATE}
```

```
WHERE USERID=#{userId,jdbcType=VARCHAR}
```

</update>

- Lister:

```
<select id="userLogIn" resultType="com.entity.User">
```

```
select * from user where USERID=#{0} and PASSWORD=#{1}
```

</select>

2.1.4 Diagramme de séquence - <<User>>

Le diagramme de séquence suivante:

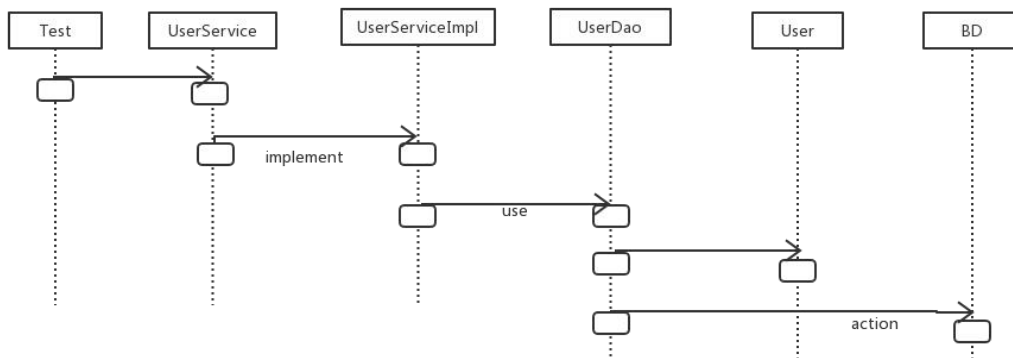


Diagramme de séquence (2-1)

2.1.5 Diagramme de classe - <<User>>

Pour bien utiliser cette classe, j' ai aussi ajouté l' autre fonctions. Donc, le diagramme de classe suivante:

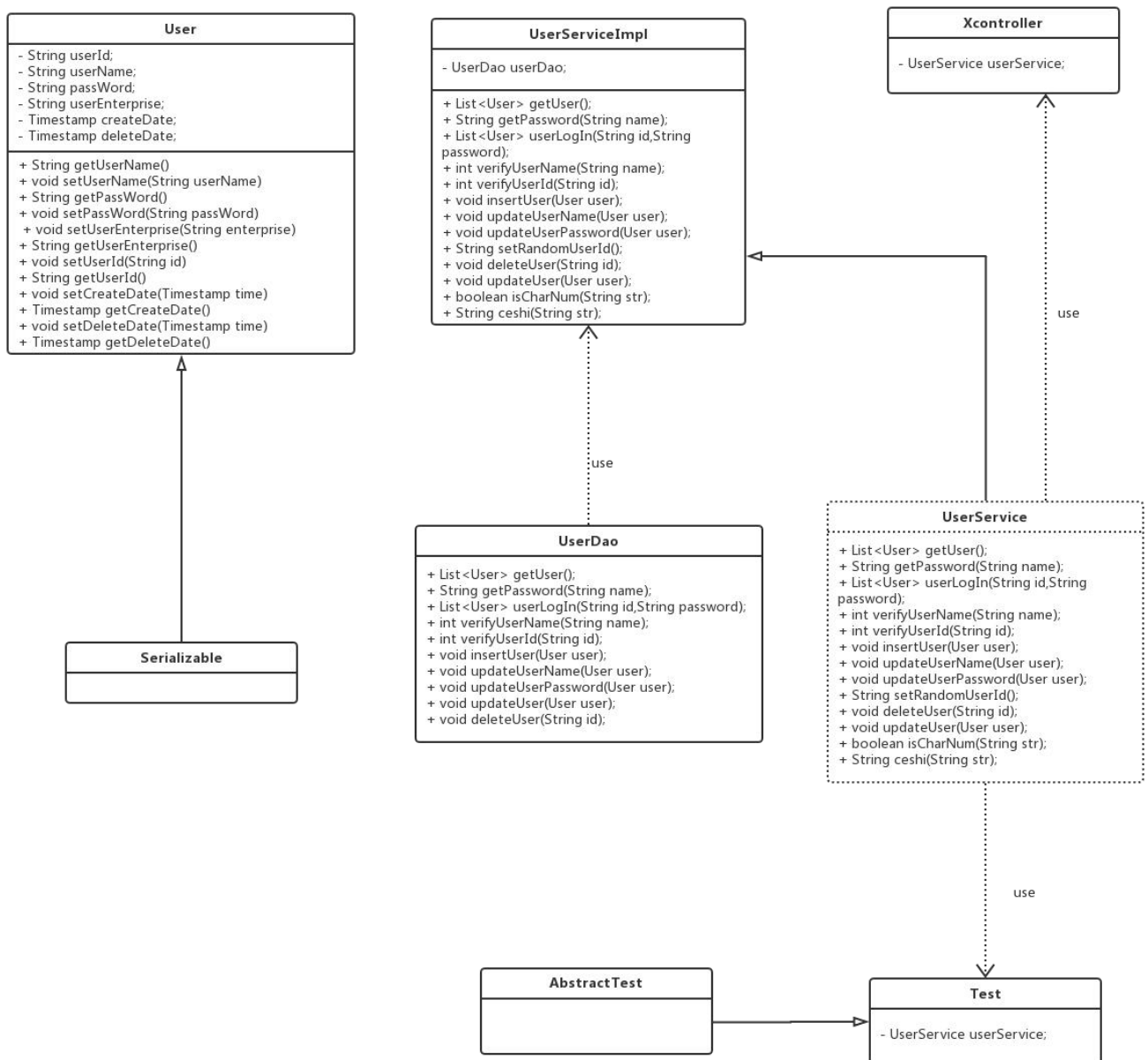


Diagramme de classe (2-2)

2.2 L ' entité - <<Book>>

2.2.1 Explication de classe <<Book>>

<< User>> peut vendre et acheter des livres en ligne. Pour la classe - <<Book>>, il y a cinq attributs:

bookId (la première clé)	ID de livre
bookName	Le nom de livre
quantity	La quantité de livre
unitprice	Le prix de livre
bookSellerId (la clé étrangère)	Le vendeur de livre

2.2.2 La classe - <<Book>>

```
public class Book implements Serializable{
    private String bookId;
    private String bookName;
    private int bookQuantity;
    private float bookUnitPrice;
    private String bookSellerId;
    public void setBookId(String bookId){
        this.bookId=bookId; //configurer ID de livre
    }
    public void setBookName(String bookName){
        this.bookName=bookName; //configurer le nom de livre
    }
    public void setBookQuantity(int bookQuantity){
        this.bookQuantity=bookQuantity; //configurer la quantité de livre
    }
    public void setBookUnitPrice(float bookUnitPrice){
        this.bookUnitPrice=bookUnitPrice; //configurer le prix de livre
    }
    public void setBookSellerId(String bookSellerId){
        this.bookSellerId=bookSellerId; //configurer le vendeur de livre
    }
    public String getBookId(){
        return bookId; //retourner ID de livre
    }
    public String getBookName(){
        return bookName; //retourner le nom de livre
    }
    public int getBookQuantity(){
        return bookQuantity; //retourner la quantité de livre
    }
    public float getBookUnitPrice(){
        return bookUnitPrice; //retourner le prix de livre
    }
    public String getBookSellerId(){
        return bookSellerId; //retourner le vendeur de livre
    }
}
```

2.2.3 Diagramme de classe - <<Book>>

Le diagramme de classe suivante:

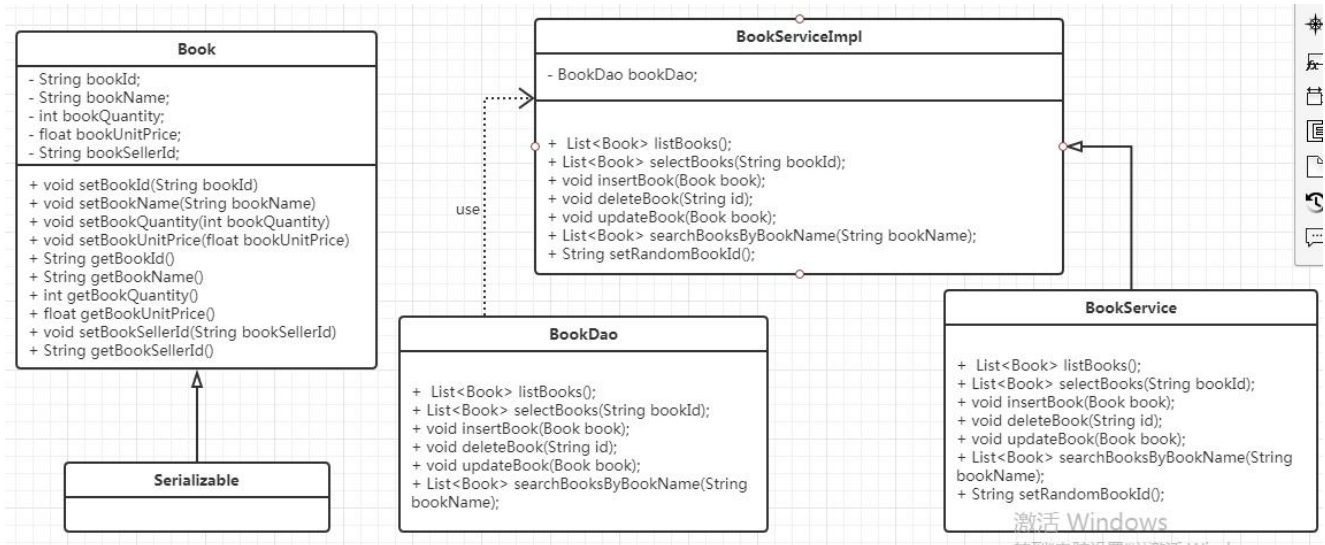


Diagramme de classe (2-3)

2.3 L'entité - <<Demande>>

2.3.1 Explication de la classe <<Demande>>

Il y a huit attributs dans la classe Demande:

demandeId (la première clé)	ID de demande
bookId (la clé étrangère)	ID de livre
quantity	La quantité
userId (la clé étrangère)	ID de client
totalPrice	Le prix de demande
payOrNot	Payer ou ne pas payer
createDate	Date de créer cette demande
deliveryAdress	L'adresse de livraison

2.3.2 La classe - <<DemandeServiceImpl>>

Cette classe réalise l'interface —<<DemandeService>>. L'algorithme suivante:

@Service //la couche de service

public class DemandeServiceImpl implements DemandeService

{

 @Resource // appeler la couche d'accès aux données

 private DemandeDao demandeDao;

 public List<Demande> listDemande(){

 return demandeDao.listDemande(); //lister tous les demandes

 }

 public String setRandomDemandeId(){

 //obtenir ID de demande


```

    }
    .....
    public String inverseDemandeToString(Demande demande){
        //obtenir tous les informations sur type de <<String>>
    }
}

```

2.3.3 Diagramme de classe - << Demande >>

Le diagramme de classe suivante:

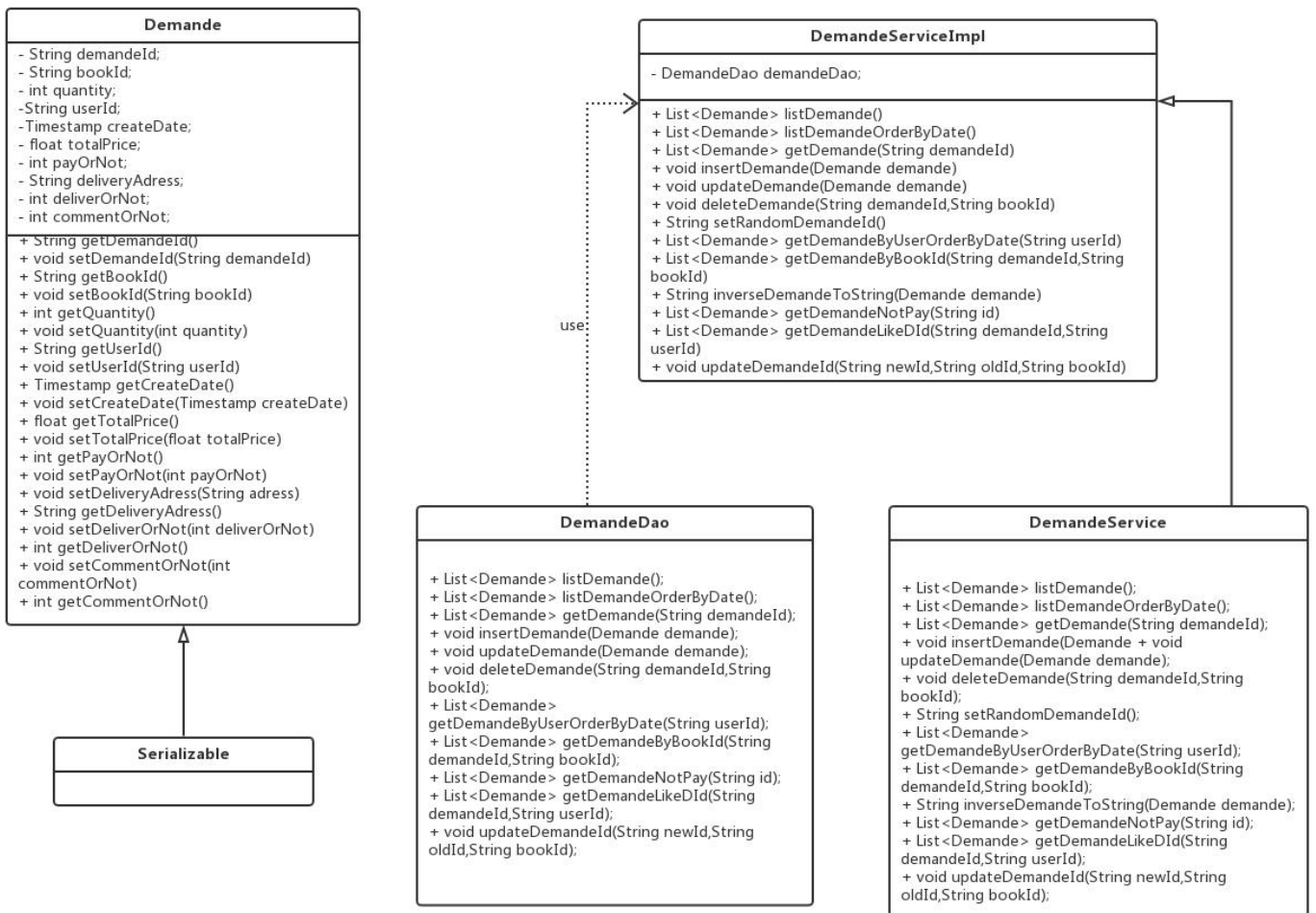


Diagramme de classe (2-4)

3. Travail (31.07.2017-25.08.2017)

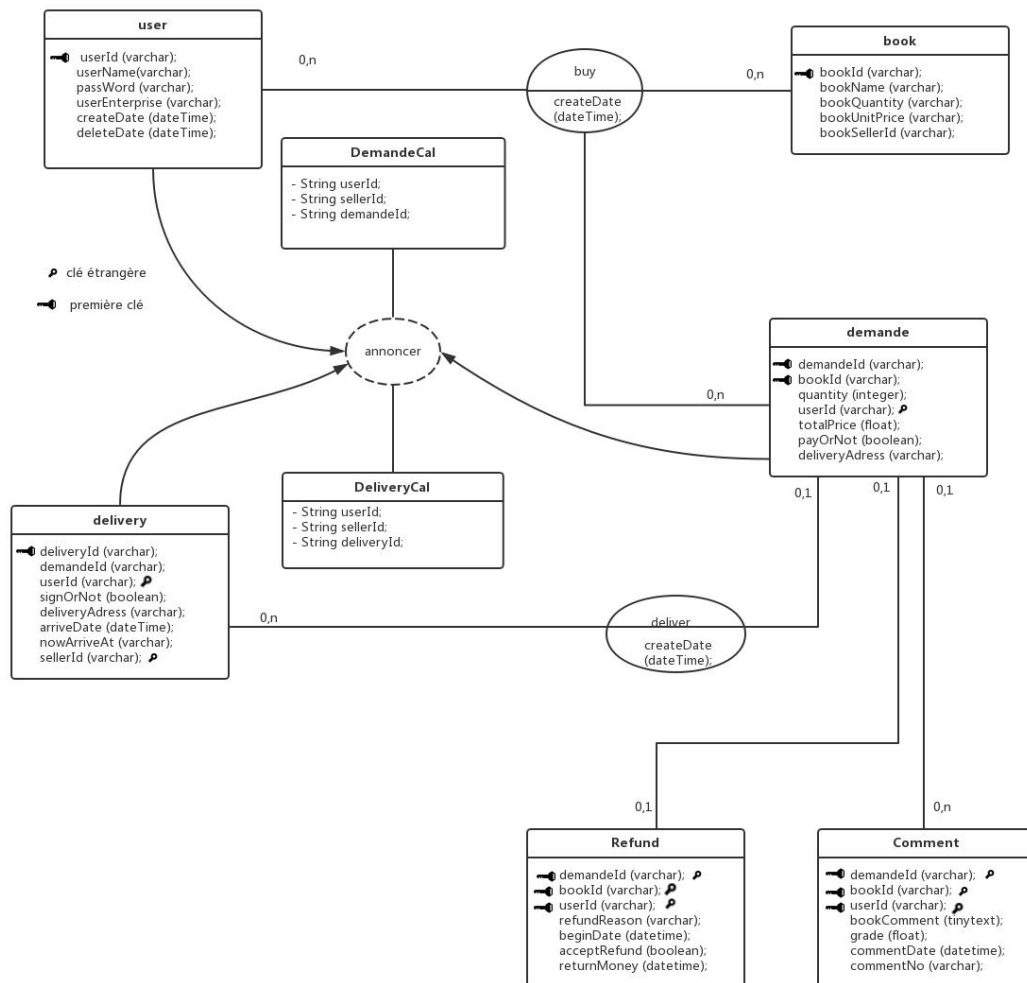
Pendant cette période, j' ai fait deux programmes:

- (1) Programme personnel —Livres en ligne
- (2) Programme d' entreprise —SAILIAN: système de GRC

3.1 Programme - Livres en ligne

3.1.1 Modèle conceptuel de données

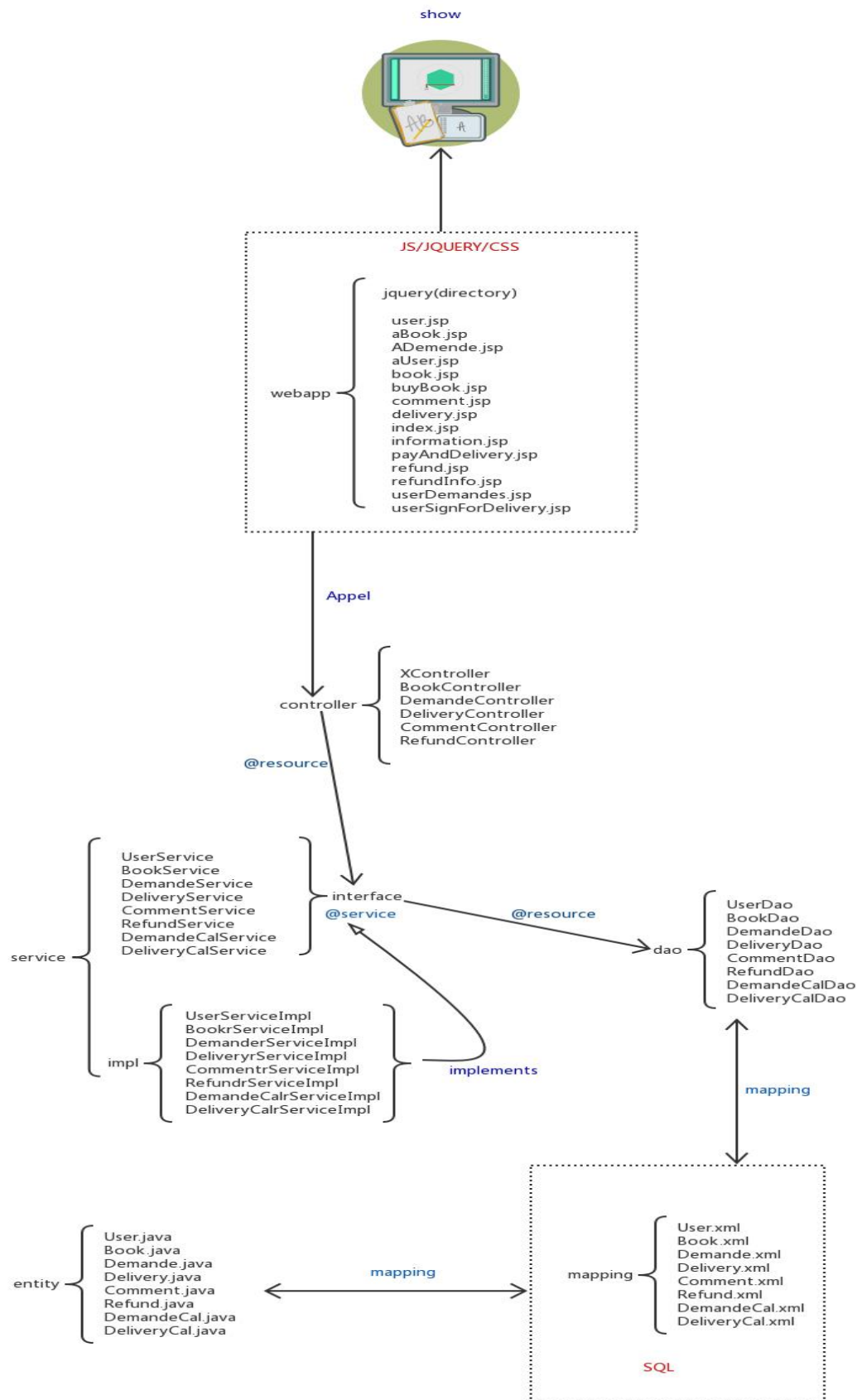
Modèle conceptuel de données suivante:



Modèle conceptuel de données (3-1)

3.1.2 La strcture de programme

La strcture de programme suivante:

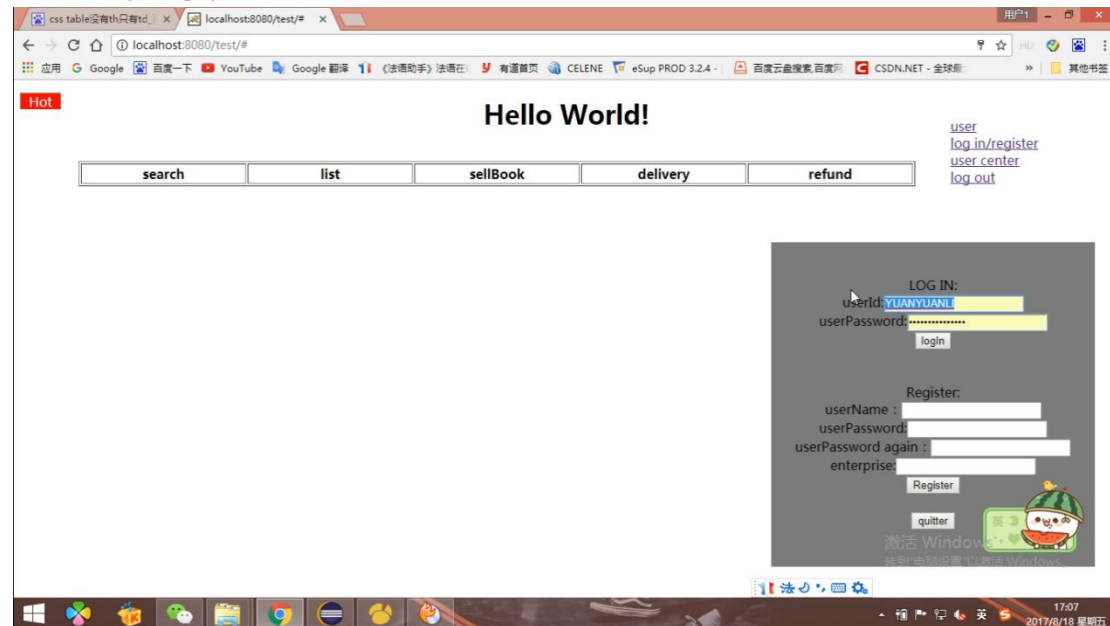


La strcture de programme (3-2)

3.1.3 Le résultat

Quelques exemples de résultat:

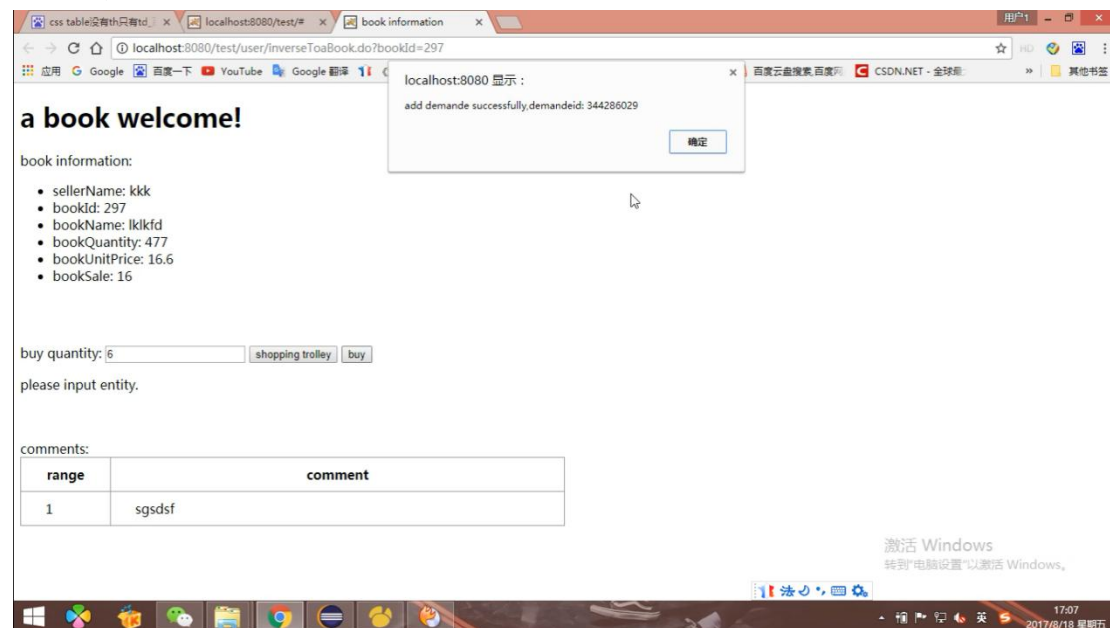
- Client Login et page d'accueil:



Login (3-3)

- Client ajoute des livres à panier ou achète des livres:

Si réussir, il va alerter ID de demande.



Acheter (3-4)

- Livraison :

Le vendeur remplit des informations de livraison et livre.

The screenshot shows a web browser window with the URL `localhost:8080/test/user/inverseDelivery.do`. The page title is "DeliveryService" and there is a "homepage" link. Below the title, there are three buttons: "waiting for delivery" (highlighted), "start to deliver", and "update delivery info".

Below the buttons, there is a section titled "1. demandes waiting for delivery : search". It contains a table with the following data:

range	demandeId	userId	bookId	quantity	cost	adress	createTime
1	344286029	24	297	6	99.6	adress	2017/8/18 下午 5:07:36

Below the table, it says "1 records".

Livraison (3-5)

- Recevoir des livres et commenter :

Le client reçoit des livres et commente:

The screenshot shows a web browser window with the URL `localhost:8080/test/user/inverseDemandes.do?userId=24`. The page title is "DeliveryService" and there is a "homepage" link. Below the title, there are three buttons: "waiting for delivery", "start to deliver", and "update delivery info".

Below the buttons, there is a section titled "4-2. search demande : demandeId: search". It contains a table with the following data:

range	demandeId	userId	bookId	quantity	totalPrice	createDate	details
1	341	24	7250	5	449.5	2017/8/18 下午5:04:42	details
2	341	24	992	6	48	2017/8/18 下午5:04:52	details
3	344286029	24	297	6	99.6	2017/8/18 下午5:07:36	details
4	35904	24	297	3	49.8	2017/8/16 下午4:46:21	details
5	364737	24	4	5	284.5	2017/8/9 下午3:32:13	deja sign details refund
6	364737	24	7250	5	449.5	2017/8/9 下午3:32:26	deja sign details refund
7	364737	24	992	6	48	2017/8/9 下午3:32:32	deja sign details refund
8	6234731606	24	7250	6	539.4	2017/8/18 下午3:52:35	deja sign details refund
9	87517200	24	297	3	49.8	2017/8/16 下午4:46:30	deja sign details refund
10	934639405	24	1763946	5	45	2017/8/9 下午3:32:21	details
11	98413827	24	992	6	48	2017/8/18 下午3:53:16	deja sign details refund

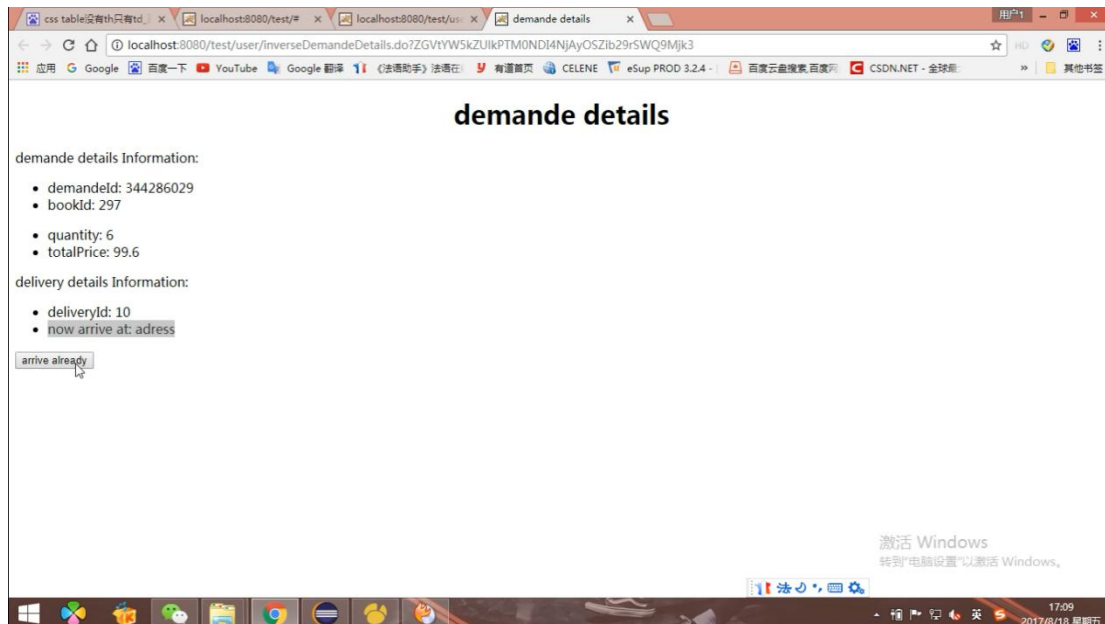
Below the table, it says "11 records".

Below the table, there is a section titled "user's demandes not pay (shopping trolley): search". It contains a table with the following data:

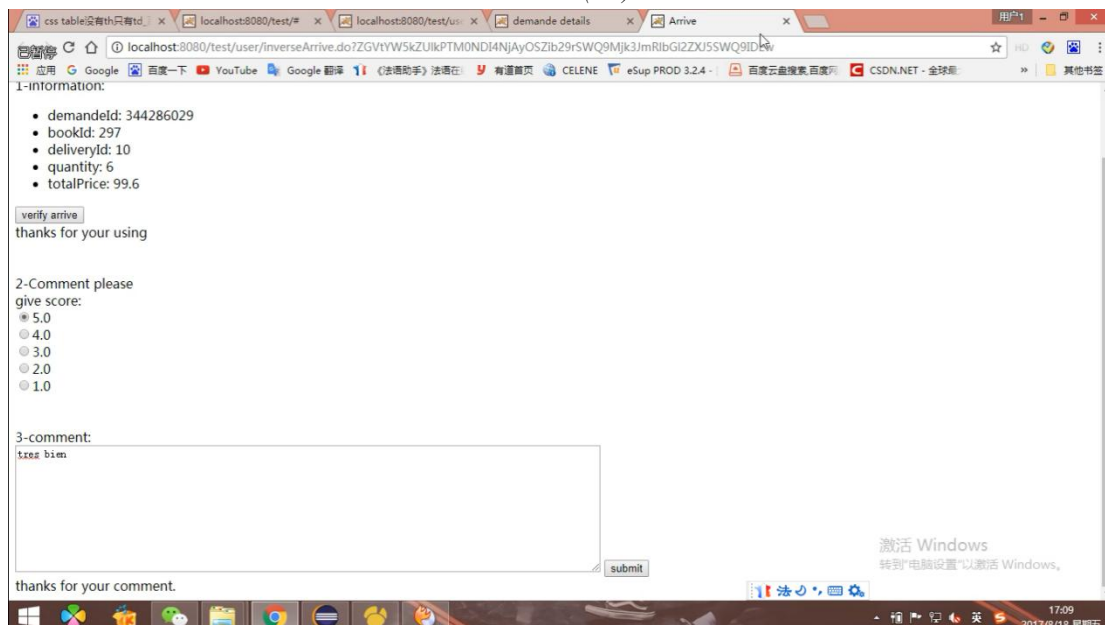
range	demandeId	userId	bookId	bookName	quantity	totalPrice	createDate	payDemande
-------	-----------	--------	--------	----------	----------	------------	------------	------------

Below the table, it says "0 records".

Tous les demandes (3-6)



Recevoir (3-7)



Commenter (3-8)

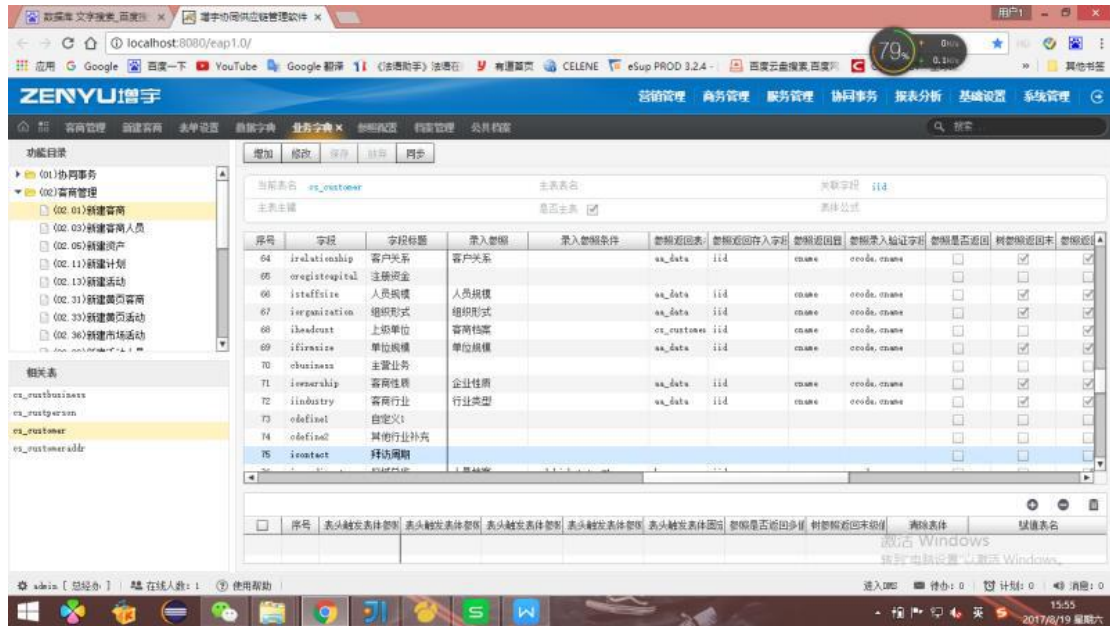
3.2 Programme —SAILIAN

3.2.1 Introduction de SAILIAN

- SAILIAN: SAILIAN est un système de gestion de la relation de client. Par exemple, on peut obtenir et modifier tous les informations de client. Dans cette système, un client correspond à une entreprise. Cette système peut être utilisée par entreprises de petites et de moyennes dimensions pour gérer ou analyser les relations de leur clients.
- L'outil: IntelliJ IDEA (jdk 1.7 + tomcat + navicat)
- La structure:
 - Flex ;
 - spring + springMVC + ibatis

La structure du programme de l'équipe CRM est très différent que le programme que j' ai fait avant.

- web:



Interface (3-9)

3.2.2 Missions

- Mettre les document correspond à les fonctions de programme sur menu.
- Réaliser fonctionnements, comme:
 - (1) Contrôle le nom de client (juste caractère Chinois, caractère anglais, chiffre, tiret bas).
 - (2) Le nom du client ne peut pas répéter.
 - (3) Lister le numéro de téléphone de contact d' entreprise.

3.2.3 Note

Quand mon stage est fini, ce programme est encore dans la période de développement.

4. Conclusion

Durant cette période de stage, je contacte deux programmes, un est mon programme personnel web de vendre livres, un est programme d'entreprise - gestion de la relation de client.

Programme -java web -vendre livres en ligne, c' est le premier java web programme personnel que je fais. Les commencements sont souvent pénibles. Au début, c' était un peu difficile pour moi pour comprendre java web, parce que je n' étudie pas de développement de java web (application) avant ce stage, sans parle que de GRC. Presque, je ne comprends pas qu' est-ce que je dois faire. Peu à peu, je pense que ça m'a beaucoup intéressant. Dans ce programme, j' ai étudié quelques annotation de java: comme @resource, @service, langage js, jquery, CSS, fais connaissances de java web. Bien que je n'étudie pas très profondément, c'était bonne acquis pour mon étude, pour moi.

Programme - SAILIAN: quand j'ai vu que les codes que j'ai écrits peuvent être utilisées, j'étais très contente, même s'il y a juste une ligne d' expression rationnelle quelquefois.

Pendant cette période, à chaque jour, sa récolte de nouveautés. Parfois, je peux juste finir un fonctionnement, mais j' ai étudié, j' ai réfléchi. Chaque jour, chacun a sa mission, j' ai fait un plan de mission de lendemain aussi, comme j' aurai programmé le fonctionnement de contrôle le numéro de téléphone. Quelquefois, il faut faire heure supplémentaire pour finir missions. Il ne faut pas remettre au lendemain ce que l' on peut faire le jour même. J' aime ce stage.

Je voudrais remercier mes collègues et l' équipe <<CRM>>, ils sont très occupés, mais quand j'ai posé questions, ils étaient très patients d'expliquer.

Mes remerciement vont également à monsieur LI Chengzhi. C' etait très gentil que Monsieur LI offre l' opportunité de faire le stage.

De même, je remercie monsieur Mathieu Delalandre. Quand j' envoie E—mail et petit rapport pour décrire la situation de mon stage, monsieur Delalandre encourage moi à bien étudier.

5. Référence

<https://baike.baidu.com/item/%E5%BE%90%E5%B7%9E%E5%B8%82%E5%85%83%E7%94%B3%E8%BD%AF%E4%BB%B6%E6%9C%89%E9%99%90%E5%85%AC%E5%8F%B8/5210879?fr=aladdin>
<https://fr.wikipedia.org/wiki/GRC>