

情報コミュニケーション学実験 II 「Web アプリケーション」レポート

立命館大学情報理工学部

氏名:ZHENG YUHUI
学籍番号:2600120572-1

2014 年 12 月 6 日

1 最終自由課題

1.1 ソフトウェア名

航空券の予約及び管理システム

1.2 外部仕様

1.2.1 管理員の方

1. login 画面の入力ボックスに管理員権限があるアカウントを入力する。

Login

User Name:

Password:

2. サービス選択ページに自分受たいサービスのボタンをクリックすると,対応の画面に入れる。

Welcome back Master!

3. 「Check the Record of all Users」というサービスを選択する時,まず,下の図のようなテーブルが現されて

Record List

Record Id	User Name	BuyTime	Plane Id	Plane Company	From	To	BoardTime
2	jack	2014-12-15 14:00:00.0	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0
3	alice	2014-12-25 11:15:00.0	1	JAL	osaka	tokyo	2015-01-20 10:20:00.0
4	alice	2015-01-05 07:25:00.0	3	MU	tokyo	shanghai	2015-01-25 16:20:00.0
6	jack	2014-12-05 20:45:08.966	3	MU	tokyo	shanghai	2014-01-25 18:30:00.0
7	jack	2014-12-05 20:46:51.188	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0
8	alice	2014-12-05 20:49:54.008	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0

6 Records found.

これは、すべての顧客の購入記録表である。 テーブルの後ろに、複合検索条件入力ボックスである

User Name:

Plane Id:

Company:

From & To

Sort by:

☐ ASC ☐ DSA

submit check

return back

ここには、四つの検索条件を指定できる: 顧客の名前, 飛行機の番号, 飛行機会社, 出発地と目的地である。もう一つ、オプションでの並べ替えも可能である(提供されるオプションは二つ: 購入時間と乗り込む時間である)後「submit check」というボタンをクリックして、検索結果のテーブルが現される。

Record List

Record Id	User Name	BuyTime	Plane Id	Plane Company	From	To	BoardTime
7	jack	2014-12-05 20:46:51.188	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0
2	jack	2014-12-15 14:00:00.0	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0

2 Records found.

User Name:

Plane Id:

Company:

From & To

Sort by:

☒ ASC ☐ DSA

submit check

return back

一方、エラー検出機能もある。検索したいレコードがなかったり、ある時入力データタイプ間違ったり、エラーメッセージも出る。

Error

Sorry for No record you want.

Error

id input error

処理した時、「return back」というボタンをクリックしてサービス選択ページに戻る。

4. 「Check or update the list of courses」というサービスを選択する時、まず、下の図のようなテーブルが現される

Course List

Course Id	Plane Id	Plane Company	From	To	BoardTime	Fare	Remaining Ticket
1	1	JAL	osaka	tokyo	2015-01-20 10:20:00.0	20000	10
2	3	MU	tokyo	shanghai	2015-01-22 15:20:00.0	30000	30
3	4	MU	shanghai	osaka	2015-01-22 18:20:00.0	15000	10
4	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0	10000	13
5	3	MU	tokyo	shanghai	2015-01-25 16:20:00.0	30000	14
6	3	MU	tokyo	shanghai	2014-01-20 18:30:00.0	30000	10
7	3	MU	tokyo	shanghai	2014-01-25 18:30:00.0	30000	29
12	3	MU	tokyo	shanghai	2014-12-25 18:30:00.0	30000	5

これは、今提供されるフライトの表である。 テーブルの下に、二つのボタンがある：「return back」というボタンをクリックしてサービス選択ページに戻る。また、「add new course」のボタンをクリックすると、新しいフライトをアペンドの入力ボックスが現される。

Add New Course

Plane Id	Board Time	Sum of Tickets		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="SUBMIT"/>	<input type="button" value="return back"/>

フライトのデータを入力して「SUBMIT」のボタンをクリックすると、アペンド処理が完了される。

Add New Course

Plane Id	Board Time	Sum of Tickets		
<input type="text" value="1"/>	<input type="text" value="2015-01-25 14:00:00"/>	<input type="text" value="50"/>	<input type="button" value="SUBMIT"/>	<input type="button" value="return back"/>

Course List

Course Id	Plane Id	Plane Company	From	To	BoardTime	Fare	Remaining Ticket
1	1	JAL	osaka	tokyo	2015-01-20 10:20:00.0	20000	10
2	3	MU	tokyo	shanghai	2015-01-22 15:20:00.0	30000	30
3	4	MU	shanghai	osaka	2015-01-22 18:20:00.0	15000	10
4	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0	10000	13
5	3	MU	tokyo	shanghai	2015-01-25 16:20:00.0	30000	14
6	3	MU	tokyo	shanghai	2014-01-20 18:30:00.0	30000	10
7	3	MU	tokyo	shanghai	2014-01-25 18:30:00.0	30000	29
12	3	MU	tokyo	shanghai	2014-12-25 18:30:00.0	30000	5
13	1	JAL	osaka	tokyo	2015-01-25 14:00:00.0	20000	50

5. 画面の左上に今のログインした客の状態コラムもあって、全部処理した時、ログアウトしてください。

User logging: admin01 (logout)

1.2.2 顧客の方

1. login 画面の入力ボックスに顧客のアカウントを入力する。

Login

User Name:

Password:

Login

2. サービス選択ページに自分受たいサービスのボタンをクリックすると,対応の画面に入れる。

Welcome here, jack!

Check your Records

Check and buy new course

3. 「Check your Records」というサービスを選択する時,まず,下の図のようなテーブルが現される

Record List of jack							
Record Id	BuyTime	Plane Id	Plane Company	From	To	BoardTime	
2	2014-12-15 14:00:00.0	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0	<button>delete</button>
6	2014-12-05 20:45:08.966	3	MU	tokyo	shanghai	2014-01-25 18:30:00.0	<button>delete</button>
7	2014-12-05 20:46:51.188	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0	<button>delete</button>

return back

これは,当顧客の購入記録表である。
毎条目の後ろに,「delete」というボタンがある。このボタンをクリックすると,削除の確認ページに入れる。

delete confirmation							
Record Id	BuyTime	Plane Id	Plane Company	From	To	BoardTime	
2	2014-12-15 14:00:00.0	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0	<button>delete</button> <button>return back</button>

赤いボタンを押すと当条目を削除する。

Record List of jack

Record Id	BuyTime	Plane Id	Plane Company	From	To	BoardTime	
6	2014-12-05 20:45:08.966	3	MU	tokyo	shanghai	2014-01-25 18:30:00.0	delete
7	2014-12-05 20:46:51.188	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0	delete

もしたくな場合は、「return back」というボタンをクリックして購入記録表のページに戻る。
 一方、エラー検出機能もある。当顧客の購入記録がない時、エラーメッセージも出る。
 全部処理した時、購入記録表の下にある「return back」という白いボタンをクリックしてサービス選択ページに戻る。

- 「Check and buy new course」というサービスを選択する時、まず、下の図のようなテーブルが現されて

Course List

Course Id	Plane Id	Plane Company	From	To	BoardTime	Fare	Remaining Ticket	
1	1	JAL	osaka	tokyo	2015-01-20 10:20:00.0	20000	10	buy
2	3	MU	tokyo	shanghai	2015-01-22 15:20:00.0	30000	30	buy
3	4	MU	shanghai	osaka	2015-01-22 18:20:00.0	15000	10	buy
4	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0	10000	13	buy
5	3	MU	tokyo	shanghai	2015-01-25 16:20:00.0	30000	14	buy
6	3	MU	tokyo	shanghai	2014-01-20 18:30:00.0	30000	10	buy
7	3	MU	tokyo	shanghai	2014-01-25 18:30:00.0	30000	29	buy
12	3	MU	tokyo	shanghai	2014-12-25 18:30:00.0	30000	5	buy
13	1	JAL	osaka	tokyo	2015-01-25 14:00:00.0	20000	50	buy

[return back](#)

これらは、今購入できるフライトの表である。
 テーブルの下に、二つのボタンがある：
 毎条目の後ろに、「buy」というボタンがある。気になるフライトに対応の購入ボタンをクリックすると、購入の確認ページに入れる。

purchase confirmation

Course Id	Plane Id	Plane Company	From	To	BoardTime	Fare	Remaining Ticket	
2	3	MU	tokyo	shanghai	2015-01-22 15:20:00.0	30000	30	buy return back

青いボタンを押すと当フライトを購入する。

Record List of jack

Record Id	BuyTime	Plane Id	Plane Company	From	To	BoardTime	
6	2014-12-05 20:45:08.966	3	MU	tokyo	shanghai	2014-01-25 18:30:00.0	delete
7	2014-12-05 20:46:51.188	2	ANA	osaka	nagoya	2015-01-12 06:20:00.0	delete
9	2014-12-07 11:43:14.43	3	MU	tokyo	shanghai	2015-01-22 15:20:00.0	delete

もしたくな場合は、「return back」というボタンをクリックして購入記録表のページに戻る。
 これまで、新航空券の購入処理が完了される。
 購入記録表画面の「return back」という白いボタンをクリックしてサービス選択ページに戻る。

5. 画面の左上に今のログインした客の状態コラムもあって、全部処理した時、ログアウトしてください。

User logging: admin01 (logout)

1.3 内部仕様

1.3.1 開発と動作環境

- プログラム言語: JAVA(servlet,jsp)
- バージョン管理: git
- ウェブサーバ: Apache-Tomcat
- 構築ツール: Apache-Ant
- データベース: H2 Database Engine
- ブラウザ: Firefox

1.3.2 クラス

- User: 全セッションの中に、各々モジュールの間でログインしたユーザのデータの通信に利用されるクラス。
- Course: フライトの表を表示して jsp からリクエストに応じるために、データベースのデータをスクラッチして通信用のクラス。
- Record: 購入記録表を表示して jsp からリクエストに応じるために、データベースのデータをスクラッチして通信用のクラス。

User	
private int	id
private String	name
private String	password
private int	authority
public int	getId()
public void	setId(int id)
public String	getName()
public void	setName(String name)
private String	getMD5(String password)
public String	getPassword()
public void	setPassword(String password)
public boolean	checkPassword(String pass)
public int	getAuthority()
public void	setAuthority(int authority)

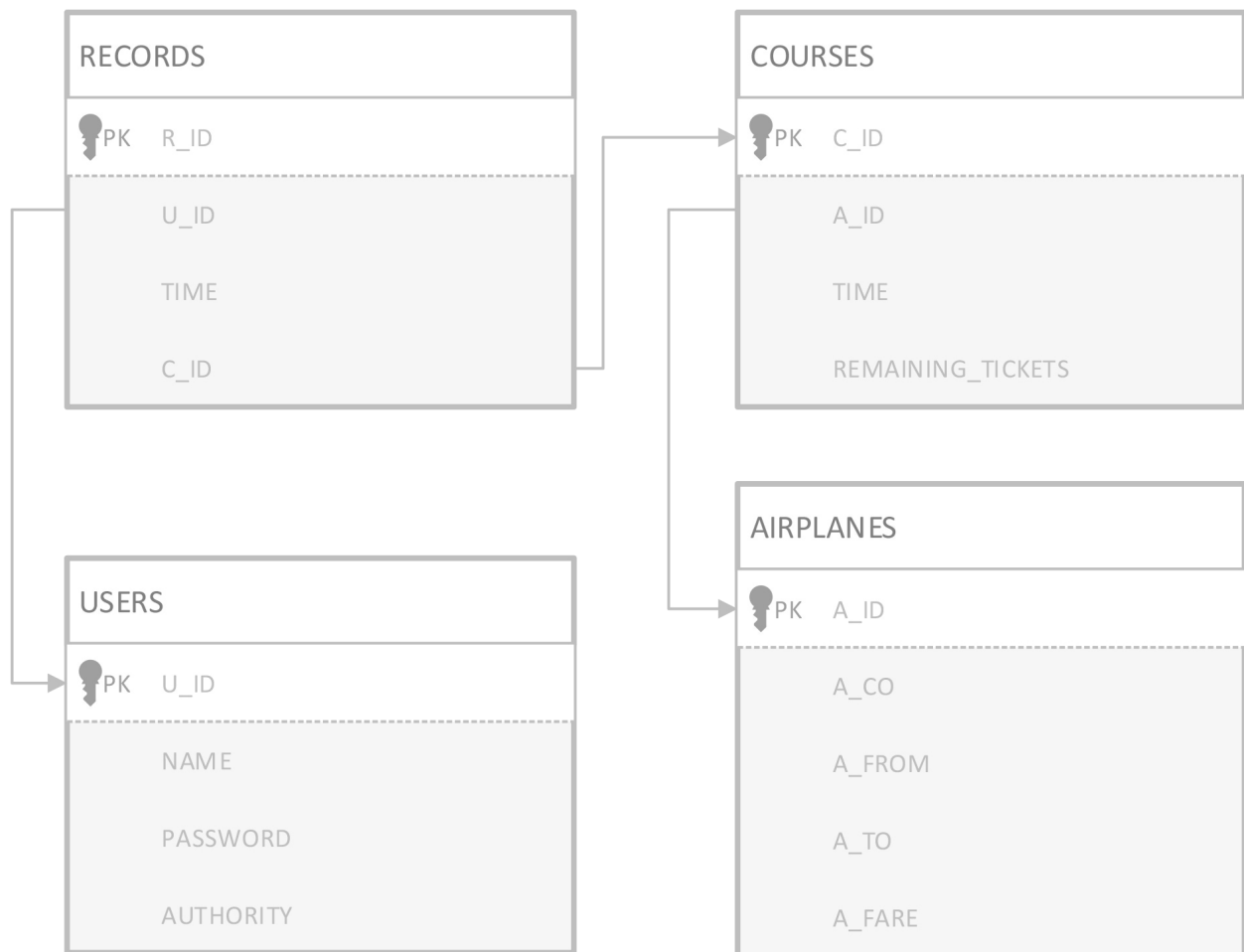
Course	
private int	courseId
private int	planeId
private String	planeCo
private String	from
private String	to
private Timestamp	boardTime
private int	fare
private int	remainingTickets
public int	getCourseId()
public void	setCourseId(int id)
public int	getPlaneId()
public void	setPlaneId(int id)
public String	getPlaneCo()
public void	setPlaneCo(String co)
public String	getFrom()
public void	setFrom(String from)
public String	getTo()
public void	setTo(String to)
public Timestamp	getBoardTime()
public void	setBoardTime(Timestamp time)
public int	getFare()
public void	setFare(int fare)
public int	getRemainingTickets()
public void	setRemainingTickets(int num)

Record	
private int	recordId
private String	userName
private int	planeId
private String	planeCo
private String	to
private String	from
private Timestamp	boardTime
private Timestamp	buyTime
public int	getRecordId()
public void	setRecordId(int id)
public String	getUserName()
public void	setUserName(String name)
public int	getPlaneId()
public void	setPlaneId(int id)
public String	getPlaneCo()
public void	setPlaneCo(String co)
public String	getFrom()
public void	setFrom(String from)
public String	getTo()
public void	setTo(String to)
public Timestamp	getBoardTime()
public void	setBoardTime(Timestamp time)
public Timestamp	getBuyTime()
public void	setBoardTime(Timestamp boardTime)

1.3.3 データベース

- AIRPLANES: 飛行機のデータを保存したテーブル。
- COURSES: フライトのデータを保存したテーブル。
- RECORDS: 購入記録のデータを保存したテーブル。
- USERS: ユーザのデータを保存したテーブル。

テーブルの内容と間の関連のことは、下の図の通り。

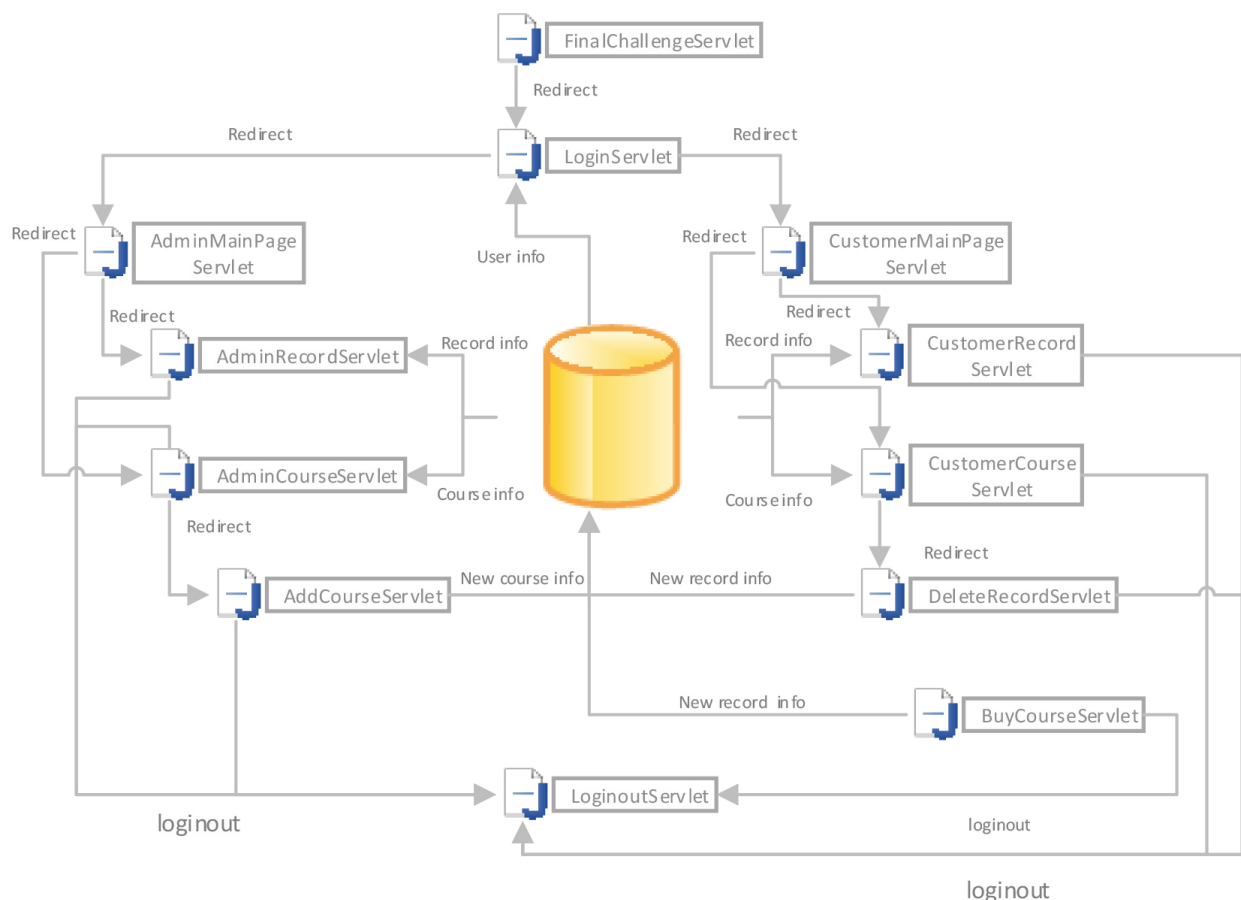


1.3.4 サーブレット

- FinalChallengeServlet: 入口として Login へリダイレクトするサーブレット。
- LoginServlet: Login リクエストの処理と、ユーザの権限により対応のページへリダイレクトするサーブレット。
- LoginoutServlet: ログインしてユーザのログアウトのリクエストを処理して、ログインページへリダイレクトするサーブレット。
- AdminMainPageServlet: ユーザ(管理員)のサービス選択により、対応のページへリダイレクトするサーブレット。
- AdminRecordServlet: データベースからすべて条件に満足した購入レコードを取り出して、AdminRecord.jsp へ転送するサーブレット。
- AdminCourseServlet: データベースからすべてフライトの条目を取り出して、AdminCourse.jsp へ転送するサーブレット。

- AddCourseServlet: AddCourse.jsp からデータを受け入れてエラー検出してデータベースに新しい条目をインプットするサーバレット。
- CustomerMainPageServlet: ユーザ(顧客)のサービス選択により,対応のページへリダイレクトするサーバレット。
- CustomerRecordServlet: データベースからすべて当ユーザの購入レコードを取り出して, CustomerRecord.jsp へ転送するサーバレット。
- CustomerCourseServlet: データベースからすべて残る航空券あるフライトの条目を取り出して, CustomerCourse.jsp へ転送するサーバレット。
- DeleteRecordServlet: DeleteRecord.jsp からデータを受け入れてエラー検出してデータベースに応じる条目を削除するサーバレット。
- BuyCourseServlet: BuyCourse.jsp からデータを受け入れてエラー検出してデータベースに新しい購入レコードを加えるサーバレット。

サーブレットの間のデータ通信と関連のことは,下の図の通り。

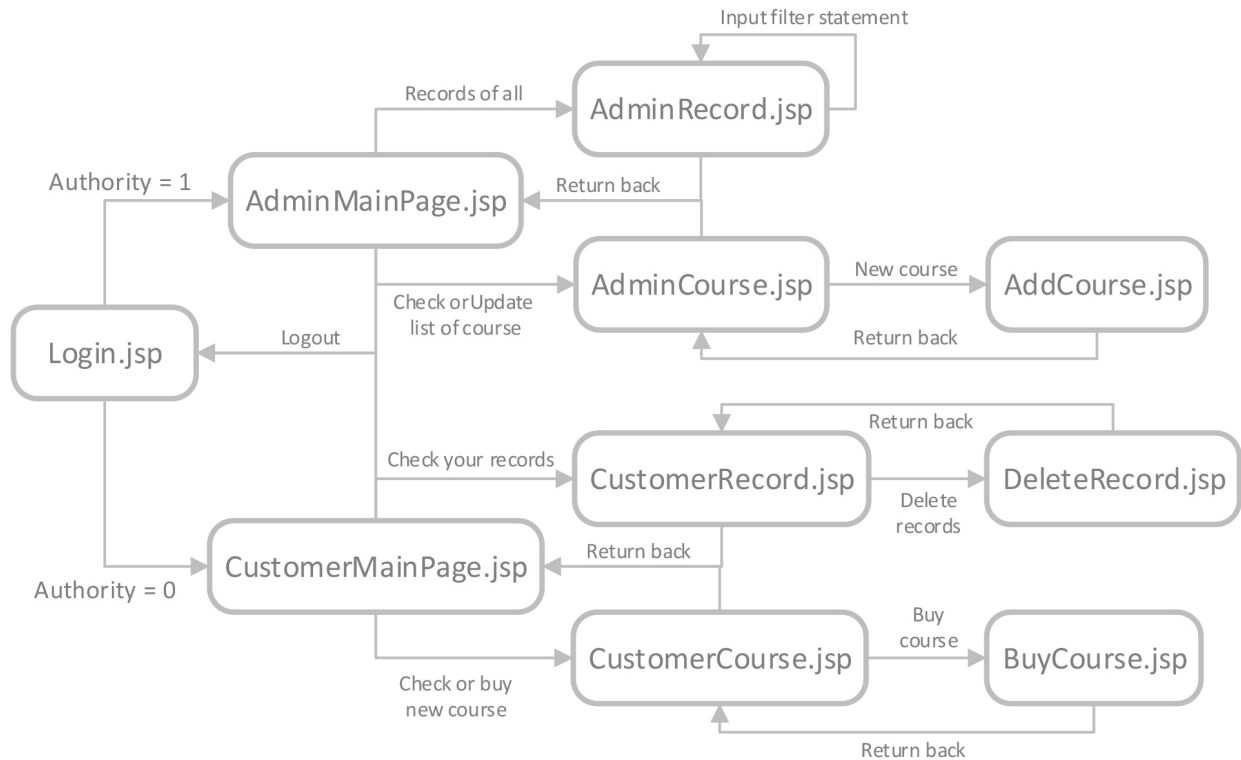


1.3.5 JSP

- Login: ログイン用のページ。
- AdminMainPage: 管理員用のサービス選択ページ。
- CustomerMainPage: 顧客用のサービス選択ページ。
- AdminRecord: 管理員が顧客の購入記録を調べ用のページ。
- CustomerRecord: 顧客が自分の購入記録をチェックして削除用のページ。
- AdminCourse: 管理員がフライトリストをチェックして新しいフライトをアペンド用のページ。

- CustomerCourse: 顧客がフライトリストを調べて航空券の購入用のページ
- AddCourse: 管理員が新しいフライトをアペンドの確認用のページ。
- BuyCourse: 顧客が航空券の購入の確認用のページ。
- DeleteRecord: 顧客が自分の購入記録を削除する時,確認用のページ。

jsp の間の関連と構成のことは,下の図の通り。



2 工夫した点,アピールしたい点

- 複数のデータベーステーブルを利用して,データの再利用可能性をアップする。
- セッションを利用して,ログインしたユーザの権限により違うサービスを提供できる。
- 一般的の入力エラー検出なども保障して,必要なエラーメッセージの理解易い提示もある。
- bootstrap という CSS フレームワーク利用して,ユーザのインタラクティビティ体験をアップする。
- git というバージョン管理ツールを利用して,エラーが出る時,rollback できる。ロバスト性と効率をアップする。
- コードレビューすることもある。冗長の部分を削除してコードフォーマットする。

3 感想・提言

技術方面は簡単だと思いきや,実際に実装したら,大変時間がかかる。一方,色々のトラブルとミスも起こした。プログラミングはただ技術と原理の理解ではなくて,根気とか真面目とか,人に対して全面要求もある。

このシステムを実装するために,いろいろ技術も参考して勉強していました。基礎的なウェブ技術について,全部 w3school[1] というサイトから習ってきまして,美化するために,bootstrap[2] という css frame も勉強していました。

コーディングのほかに,バージョン管理には github[3] でオンライン保存で,レポートを作るため,オンライン LaTeX サービス sharelatex[4] も利用する。

もう一つ,自由課題の選択に困ることもある。最後,この「航空券の予約及び管理システム」を決めた理由というのは,データベースの利用できることはともかく,最近私ら中国からの留学生が,春休みの時帰国につて喋っている。基本的なニーズがあれば,このシステムを深く理解したくなる。自分で簡単な「航空券の予約及び管理システム」を実装した

ほうがいいじゃないでしょうか,と思うなら,これは前提として,最後このテーマに決めた。興味を持っているから,トラブルを起こした時,落ち込みにならずに,一生懸命頑張って,目標の80%までできました。もし興味もないし,ランダムな課題を作れば,今のようなプログラムができることは想像もできないと思う。

ですから,いろいろ不足もあるのに,この「航空券の予約及び管理システム」がただ宿題ではなくて,私の自慢の作品と言い過ぎることがありません。

お楽しみください。

References

- [1] w3school,<http://www.w3schools.com/>, (2014/12/06).
- [2] bootstrap,<http://getbootstrap.com/css/>, (2014/12/06).
- [3] github,<https://github.com/>, (2014/12/06).
- [4] sharelatex,<https://sharelatex.com/>, (2014/12/06).