Job Portal

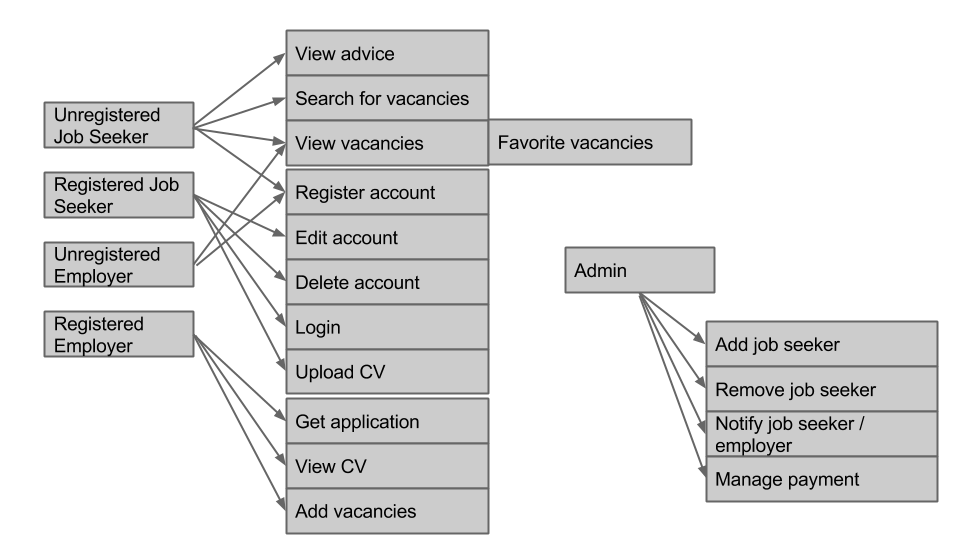
Team members: Chih-Chuan Chan,

Chong Yue,

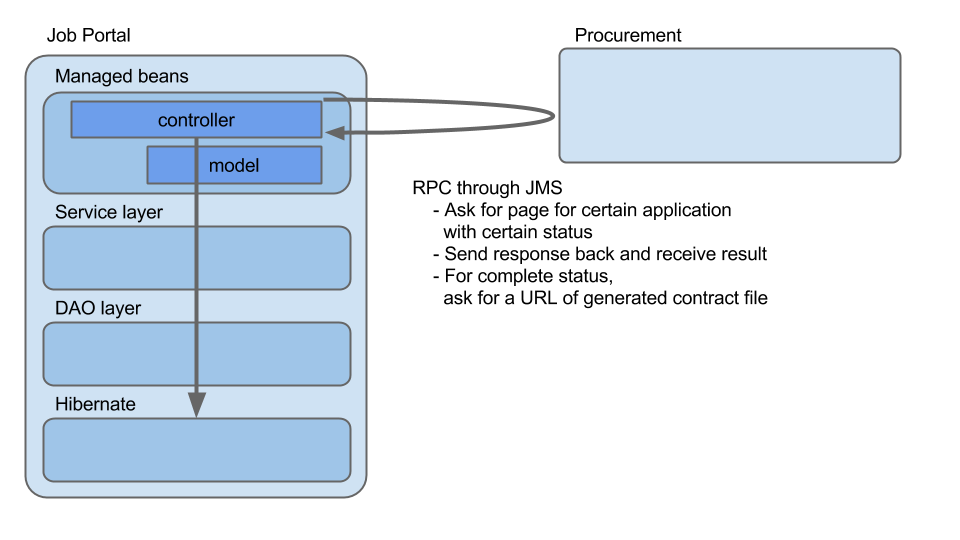
Cheng Yang

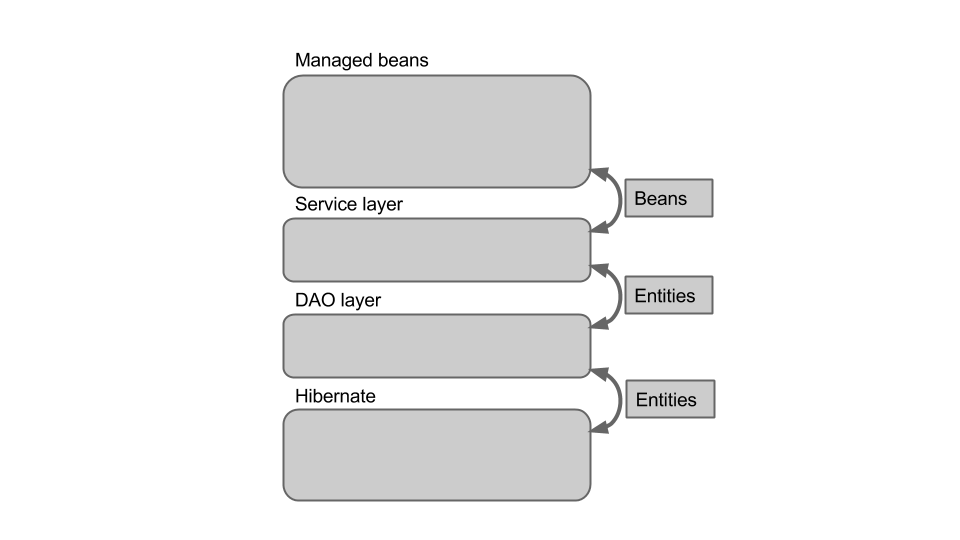
Last updated: 3/24/2014

**Feature description**

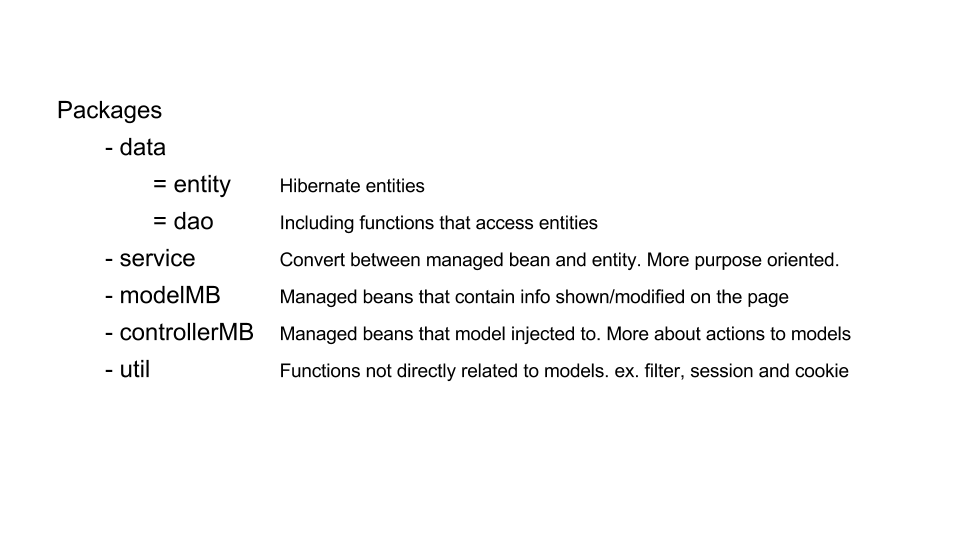


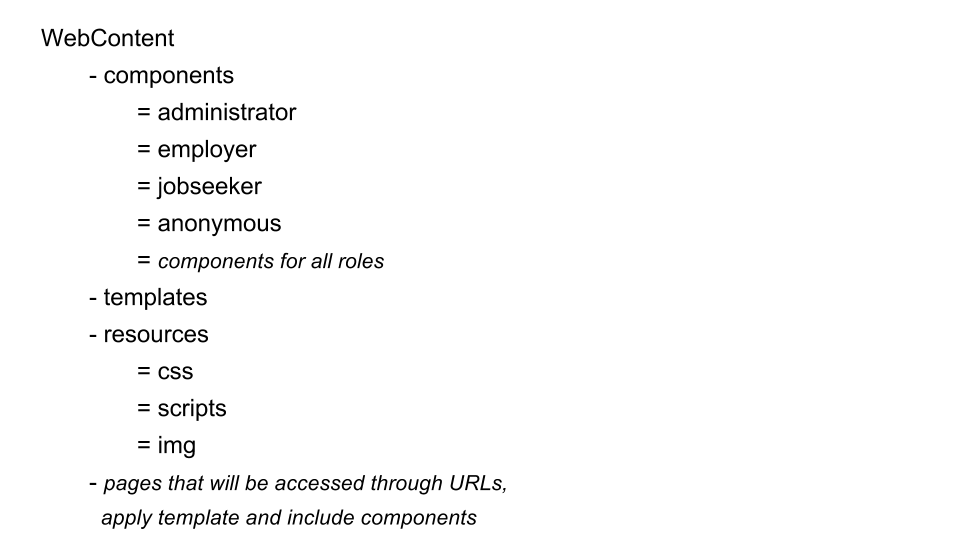
**Architecture**



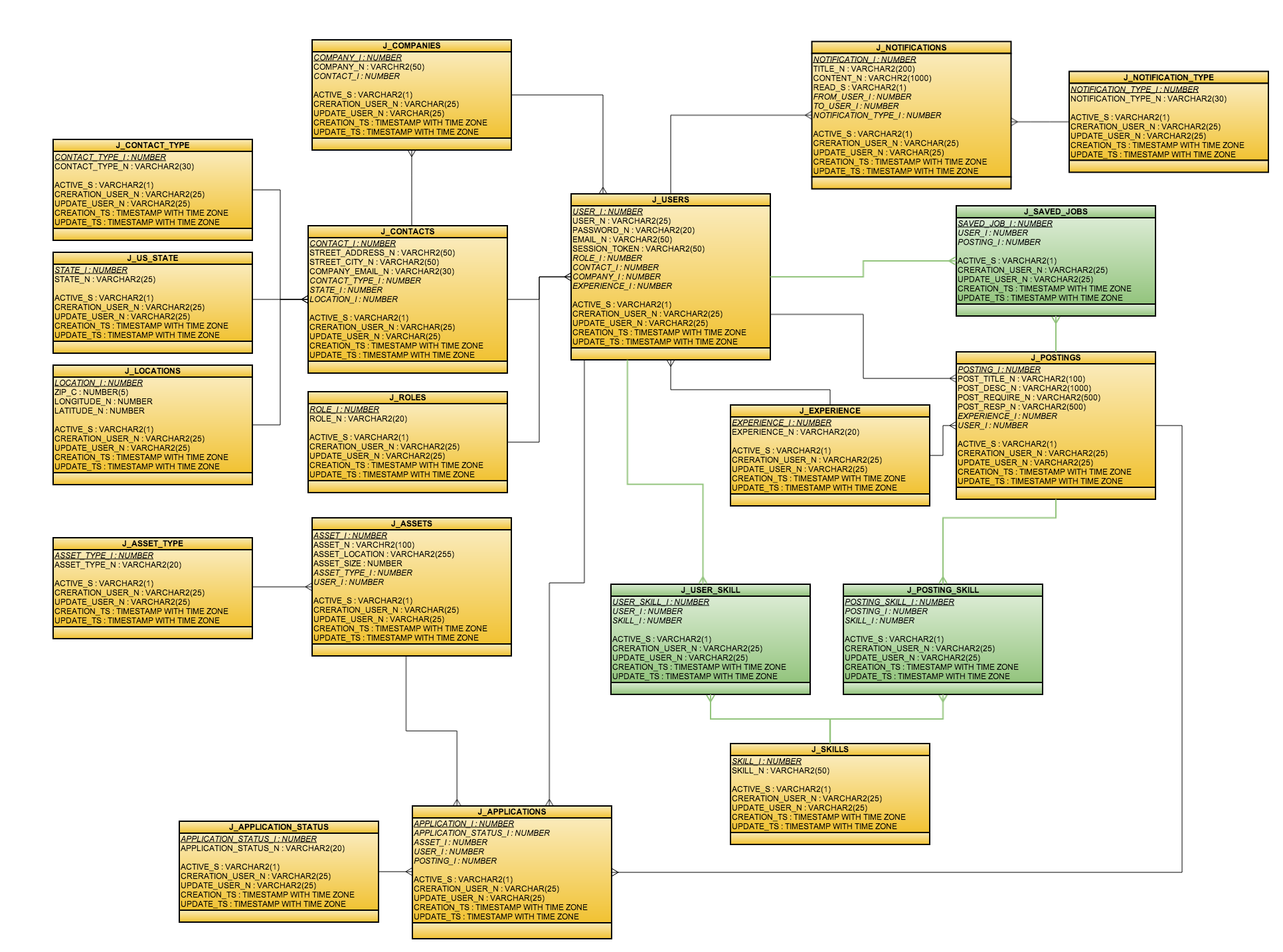


**File structure**





**Database schema**



**Beans**:

1. Job

2. User

- Job seeker

- Employer

3. CV

4. Transaction

5. Notification

**JSF actions:**

X1. Login / logout

chong 2. Create / delete account

chong 3. Create / block account (admin)

cchan 4. Add / remove job vacancies

cchan 5. Add / remove job vacancies (admin)

=6. Create / remove transaction

=7. Log list view

cchan 8. - Job vacancy list view

cchan - Job view

9. - CV list view

- CV view

- Application list view

10. - Job seeker list view

- Job seeker view

11. - Employer list view

- Employer view

12. Favorite job list view

=13. - Transaction list view

- Transaction view

14. Job vacancy history

**JSF AJAX actions:**

1. Edit account / upload CV (10.2)

2. Search job position (8.1)

3. Search CV (9.1)

4. Edit job vacancies (8.2)

5. Edit transaction (13.2)

6. Search log (7)

**Session and Cookie:**

1. User information

2. Favorite list

3. Recently search

**Notification service:**

1. Email

- Auto-send

2. On page

- Create notification

- Notification list

- Notification detail

**Permission and role:**

1. Administrator

2. Employer

3. Job seeker

4. Anonymous

Upload / download file

Single sign on

**DAO** (\* fixed table)

1. User

- login

- logout

- getUserById

- getUserByEmail

- createUser

- updateUser

- blockUser

- listAssets

- listApplications

- search(exp, skills)

- listNotifications

2. Role\*

- listAll

3. Contacts

-

4. Contact type\*

- listAll

5. US state\*

- listAll

6. Location

-

7. Notification

- createNotification

- listNotificationByFromUser

- listNotificationByToUser

- deleteNotification

8. Notification type\*

- listAll

9. Experience\*

- listAll

10. Posting

- listJobs

- search

- createJob

- UpdateJob

11. Skill

- createSkill

- getSimilar

12. Asset

- createAsset

- updateAsset

- searchAssetByUser

13. Asset type\*

- listAll

14. Application

- createApplication

- updateApplication

- search

15. Application status\*

- listAll

|  |  |  |
| --- | --- | --- |
| # | Web page components | Service function |
| 1 | Header (login, greeting) | User.login |
| 2 | Register page | User.createUser  Roles.listAll  States.listAll  ContactType.listAll |
| 3 | Edit user page | User.updateUser  Skill.createSkill |
| 4 | User description | User.getUserById  User.getUserByEmail  User.getUserByName |
| 5 | User search list | User.search(skill, exp) |
| 6 | Create job page | Job.createJob  Skill.createSkill  Experience.listAll |
| 7 | Edit job page | Job.updateJob  Skill.createSkill  Experience.listAll |
| 8 | Job description | Job.searchJobById  Job.apply  Job.inactivate  Job.listApplicants (employer only) |
| 9 | Job search list | Job.search |
| 10 | Saved job list | User.listSavedJob |
| 11 | Create asset | Asset.createAsset  AssetType.listAll |
| 12 | Edit asset | Asset.updateAsset  AssetType.listAll |
| 13 | Asset description | Asset.getAssetById |
| 14 | Asset list | User.listAssets |
| 15 | Create application page | Application.createApp |
| 16 | Application description | Application.searchAppById |
| 17 | Application list (job seeker) | User.listApplications |
| 18 | Create Notification page | Notification.createNotif  NotificationType.listAll |
| 19 | Notification description | Notification.searchNotifById |
| 20 | Notification list | User.listNotifByFrom  User.listNotifByTo |
| 21 | Posting list (employer) | User.listPostings |
|  |  |  |

**Integration with Application service**

JMS

Create contract (4)

. div pop-up

. job vacancy -- general contract

Apply -- notify (9.2)

. div pop-up

. Questionnaire

. Bid

Give Offer -- notify (9.2, 9.3)

. Another page

. bid -- application

. finalized contract

Accept / deny -- notify (9.2)

**Application states:**

. applied

. offered

. rejected

. accepted

. denied