Design Specification (Database , UI and Servlets)

for

Java Project - Project Management System

Prepared by:

Prateek Chandan - 120050042
Anurag Shirolkar - 120050003
JVS Shyam - 120050052

CSE , 3rd Year
IIT Bombay

Under Prof. N.L. Sarda
Department of Computer Science
IIT Bombay

Date: 30th October , 2014

Table of Contents

- Database Design Specifications
  - Entities in Database
  - Relations in Database
  - ER MODEL
  - Normalization Report
- User Interface Design Specifications
  - List of Pages for user Interface for Viewing
  - List of Pages for user Interface for Editing
- Servlet Design Specification (Package and Classes)
  - DB Package
    - Connection Class
  - Project Package
    - Add Class
    - All Class
    - ApplyProject Class
    - ChangeRequirement Class
    - ChangeStatus Class
    - EditProject Class
    - SaveRemark Class
    - Search Class
    - TagAdd Class
    - Tags Class
    - View Class
  - User Package
    - Login Class
    - Logout Class
    - EditProfile Class
    - Profile Class
    - Signup Class
- Other Design Details
  - Templating Using JSP
  - External API's for UI

I. Database Design Specifications

Entities in Database

The database consists of **3 main entities** which are :

a) **USER** - userid as primary key - This table consists of Users with their personal details as name, email , roll no , phone no , password , phone no , Institute name and what kind of user they are (student , professor , alumni , others)
Other constraint on table :
check on usertype in ('student' , 'professor' , 'alumni' and 'others')
userid as primary key because it will uniquely identify the user

b) **Projects** - project_id as primary key - This table will contain details of all the projects which are added in the project database. This will have a project_id , name of project , added by (userid of user adding the project) , project status
Constraints :
addedby is a foreign key referencing userid on user table
check on project status in ('completed' , 'in progress' , 'hold')
project_id will uniquely identify the project so it is primary key

c) **Tags** - tagname as primary key - This table will have just one primary key as tagname. This will tell what all skills a user can have also what all tags(skills) a project is using
There is only one primary key which is the tagname itself

Relations in Database

For these three entities there are three relations defined on them as follows :

a) **Skills -** This is a relation on user table and tags as it will have a userid and a tagname defining what skills user have

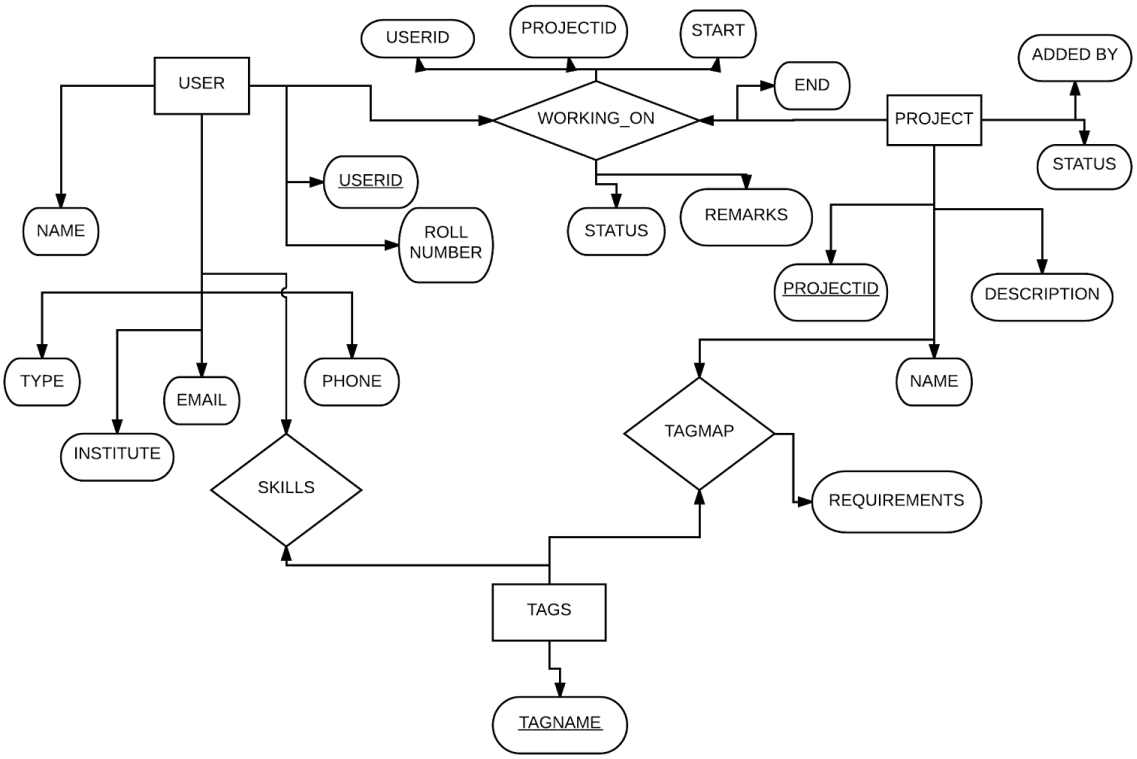
A primary key on (userid and tagname) as one user can have on skill only once

b) **Tagmap** - This will be a relation on users and projects. This will also contain no of users required for working on that project with that particular skill.

A primary key on (project_id and tagname) as one project can have on tag only once

c) **Working_on** - This will be a relations and projects as it will define which user is working on which project. It will also have a status (working status) , remarks by the person added that project to the user , start time and end time
Constraint :
check on status in ('applied' , 'working' , 'hold' , 'completed')
primary key on (userid and project_id)

ER MODEL



Note : In the above ER Model , the arrows are not proper and has to be ignored due to the constraints in software used in designing ER Model

Normalization Report

The Database is designed with proper normalizations as all the redundancy in tables and its fields are properly removed. As to store the skills of user we are using a different table instead of storing all the skills in same table and we moved it to a different table and similarly with user working status and project tags

II. User Interface Design Specifications

List of Pages for user Interface for Viewing

The following pages will be used as user interfaces for viewing in our project

- ➔ **Home Page** (URL : /) : This will have a basic search button on the front to facilitate user to search among multiple projects also having detailed descriptions of features and instructions , utilities and quick instructions of how to use the project
- ➔ **All projects page** (URL : / projects) : This will be page which will list down all the projects (relevant projects if no of project is large) with the project description and

also the search bar and list of tags on the side of it so that the users can browse projects by tags or search the projects

- ➔ **Project Specific Page** (URL : /project/project_id) : This page will contain details of projects with project_id as int the URL. The details include the Project name , the user who added this project , project description , Users who are working on this project with their remarks and working status also the page will have all the tags with requirements on this page
- ➔ **Login Page** (URL : /login) : This page will contain a form where user can fill in his login credentials to login into the system
- ➔ **User Profile Page** (URL : /user/userid) : This page will be homepage of a user. It will contain all the user details as his personal info , skills , the projects added by the user and the projects he is working on with working status and remarks
- ➔ **Tag Page** (URL : /tags/tagname) : This page will contain all the projects related to that particular tag along with the project description and the user who added the project
- ➔ **Search result Page** (URL : /search?q=string) : This page will display all the projects related the the search query “string” searching from the project name , description and list of tags and will display the related projects as their name , description and the person who added the project

List of Pages for user Interface for Editing

The following pages will be used as user interfaces for editing data in our project

- ➔ **Sign Up Page** (URL : /signup) : This page will contain the a form where user can fill in his details to add himself as a user in the system
- ➔ **Page to add project** (URL : /add-project) : This project will contain the fields to add project name , description in a proper WYIWYG Editor (What you see is what you get) , and tags. The user can then save this project
- ➔ **Page to edit project** (URL : /edit-project/project_id) : this page will open all the data related to that project in the editor and user can then edit them and save the project
- ➔ **Page to apply the project and accept users and remark them** (URL : /project/project_id) : This page will be same as the project display page. But when logged in will display a button for use to apply in that project. When admin opens the page he will get options to accept/delete and change status of user , edit page and skills for that project
- ➔ **Page to edit-profile :** (URL : /edit-profile) : This page will have all the data of user previously added and here he can edit his details . This is accessible only after login
- ➔ **Page to add new tag** (URL : /add-tag) : This page will contain a field for user to add a new tag

III. Servlet Design Specification (Package and Classes)

DB Package

This package will handle all things which is related to database connectivity

Connection Class

This class has a connection object and all the credentials to connect to the database. The object of this object is used in all other classes where connection to database is used. This is done to prevent entering database credentials at multiple places

Project Package

This package contains all the servlets as classes used for all the project and tag related works in the system
The Servlets created are as :

Add Class

This servlet on get request will display the page to add new project with all other fields . While on Post request it will check if all the fields entered by user is correct and then it will add the new project to the database

All Class

This is a simple Servlet which will fetch all the projects from database and then pass it to the jsp page to display them to the user on both get and post request

ApplyProject Class

This project takes the project id and user id from the request and adds user as to be working on that project after checking if that user is actually logged in into the system

ChangeRequirement Class

This servlet takes in the project_id , tagname and the new changed requirement as input and then modifies the people required in that tag and project. It throws an error if any other user than admin try to access it or even if the tag and project id are different

ChangeStatus Class

This Servlet changes the working status of a user on a particular project. It will change status only when the admin of that project sends request.

EditProject Class

This servlet takes in input of the project_id and the modified field inputs and then update the project details as the modified version

SaveRemark Class

This servlet takes in the input of the user id and remark and project and modified the remarks of that user working on that project only if the request is sent by the admin

Search Class

This servlet takes in input a search query and and searches that query into the database to generate the most close matching projects and send it to the project display project to display it

TagAdd Class

This servlets takes in input of a new tagname and then adds it to the database. This addition is done only if user is logged in. On get request it just display page with field to add new tag

Tags Class

This project takes in a tag as input from the url and then search for all the projects belonging to that tag and then sends it to the display page to display the projects

View Class

This class will take in a project id and display the page related to that project with all the user working on that project , project details and other skills related to that project. It also check if the user is admin then it will display all admin options and if the user is logged in and not related to that project then it will display an apply button on that project

User Package

This package contains all the the servlets as classes required for user management. The servlets are as follows :

Login Class

This servlet on get request displays user login page with the login field inputs while on post request takes in the login fields as parameter and validates if the user is logged in .

Logout Class

This servlet deletes all the session variables and makes the user to logout

EditProfile Class

This servlet takes in input the user id and on get request display a page with all the user details on it where the user can edit those details. Upon post request it saves all those edited user detail

Profile Class

This servlet takes in userid as input and displays the page related to that user. The page also displays what are the skills of the user , on what projects he is working on with project status , and what are the projects added by him

Signup Class

This servlet on a get request displays a page where a user can input various fields to signuo into the system and on post request it takes in those request and the adds the new user to the database

IV. Other Design Details

Templating Using JSP

For the user interface design , we will implement the a templating system using the jsp. This will be done by defining a separate header and footer pages which will include all the dependencies like CSS , JavaScript etc on the page and need not to include it again and again on every page.
To keep those JSP files unreachable from the URL we will keep all these templates in WEB-INF folder . This fill facilitate in fast and clean UI implementation

External API's for UI

1. For making the UI beautiful we have used a front end platform named as **Bootstrap (**http://getbootstrap.com/) . It contains pre written CSS and javascripts for beautiful UI Implementation
2. **JQuery:** Its a javascript framework which helps in writing Javascript more fastly and efficiently with less effort and much functionality
3. Other JQuery API's like
   a. Chosen JQuery : To make multiple select buttons more elegant and beautiful
   b. Redit.js : This is an API which provides the functionality of a WYSIWYG Editor into web pages
   c. PrettyPhoto JQuery API : This facilitates into beautiful display of images used in the project