



Home ► My courses ► Miscellaneous ► jdi ► Multithreading, Collections, Generics ►

Mini Project: Home Job Marketplace

## Mini Project: Home Job Marketplace

**Mini project:** Idea is to create a program which emulates the sitter/seeker enrollment

**Description:** Create a java program to create enrollment for member with details like name, address, phone no.... take input from user for the different fields and save it in db using jdbc


### Note:

- > Before starting the project please make sure to finish the Good Coding course and also make sure you understand all the normalization principles of DB as well.
- > For console Based project i.e. 1st project : can show and use Id's as inputs for any operation. Here id's means the primary key's of each table.
- > For UI Based project i.e. 2nd and 3rd Projects: Don't show id's to the user (use it internally as hidden fields alone).
- > For all tables add Status (ACTIVE,INACTIVE), When Created, When Modified columns.
- > Make sure to have all the validations for all the values you are taking from user i.e. both on java layer as well as on database constraints as well.
- > For UI Based Projects, in case of validation errors, show the error messages specific to each field next to it and also user entered valued should be pre filled as well.

**Description:** There can be two types of Members: A Seeker, who can post a job and A Sitter, who can apply for the job.

The details which are common to both seeker and a sitter have to be stored in Member table and the details specific to seeker or sitter have to stored in there specific tables.

A Member will have the following fields:

Field Name	Type
Id	Auto increment int (32)
First Name	var char (32)
Last Name	var char (32)
Phone Number	Int(10)
Email Address	var char 100
Type	Enum (Seeker, sitter) --Use enum in both DB level as well on java level
Address	var char 256 

The Email Address and Member ID has to be unique. One email can be associated with only one member.

The Seeker will extend Member and have the following extra field:

Member ID

Number of Children in Family(optional)

Spouse name(optional)

The Sitter will extend Member and have the following extra field:

Member ID

Years of experience

Expected pay

A Job will have the following fields:

ID

Title

Posted By(Member ID)

Start Date Time (DateTime)

End Date Time (DateTime)

Pay per hour (should accept decimal values)

A Sitter can apply for a job. The Application will have the following fields:

ID

Job ID

Member ID

Expected pay

Design a schema for this model. Using jdbc, create a java project to simulate the above model, storing all data in a database.

### Requirements about the user operations or flows is as below:

#### Registration flow:

All the required fields for creating a seeker and sitter have to collected in a single form.

Once user chooses to become a seeker/sitter only after that show the fields specific to that member type in the same page.

After registering user should go to his corresponding accounts pages where he will be allowed to do his member type specific operations. Like for seeker posta job, viewing applicants and stuff like that and for sitter, searching for jobs, applying for jobs and stuff like that.

#### Seeker flows:

1) Post job.



2) List job which are posted by by seeker alone. (show title, status, start date, end date), should have edit job, list applications and close job options as well.

2.1) on console provide these operation as menu inside list job.

2.2) for UI based, make all the operations like edit job or list applicants or close job as button/links in a actions column or so of the job list table itself.

3) List Application



Should show Job Title as header of the page.

show table contains column with Sitter name, Application Status(active/inactive), Expected pay.

Allow user to edit all the job fields.

#### 5) Close Job

Mark status as inactive and don't show it in the list further.

#### 6) Close Account

Prompt for confirmation to check if they really want to close account. After confirmation,

We should mark the status of member as inactive and also close all the jobs related to that member as well. And then take to a visitor page where we show confirmation page that user account is closed

#### **Sitter flows:**

##### > List all Active jobs

Show all the Active jobs which are not yet applied by sitter.

The list table show have job title, Pay per hour, Start Date time columns and Apply option for UI based project

##### > Apply job page

Once user clicks on apply link on job list page we user will land in this page and here we should be collecting Expected Pay field from sitter.

After apply the job user should be taken to list of applications page.

##### > List all applications



Here we should list the job's which user has applied. Required Columns to be shown are : job title, Expected Pay, Pay per hour, Status and Action (action will have option to delete an application)

##### > Delete an application

When user opts of delete an application. Please ask them a alert confirmation if they really want to delete the application.

Once user confirms inactive the application and take the user back to list all applications with a confirmation message saying the application was deleted.

##### > Close Account

Prompt for confirmation to check if they really want to close account. After confirmation,

We should mark the status of member as inactive and also close all the job applications related to that member as well. And then take to a visitor page where we show confirmation page that user account is closed

#### **Common flows for Both seeker and sitter;**

##### > Edit user information

Should show all the values which user entered at registration time and allow him to edit those feels and based on user who is logged-in we should show only the fields related to him.

##### > Allow search members based on partial email address.

Have a search page where users can search members based on email address and if any match occurs we can show those member's in the result page.

Interested values of each members to be displayed in results page are: name, email, member type, phone number.

##### > Forgot password

If user forgets password we can send a mail to user which contains a link to reset the password.

=====

**Important points:**

Understand the requirements thoroughly.

List out all the questions if any and discuss with mentor.

Prepare a design and present it to the mentor.

Before starting to code please make sure you get the design reviewed by mentor and make sure you are clear with the approach.

Finish the code and make sure you have one internal demo with your mentor and implement all the comments he has given

Upload the completed assignment into moodle as a zip file.

(Can invite more dev's to the presentation)

Since we have 3 projects which have same requirement but the technologies will differ, we should make sure that code which is written in first project should be reused and modularised so that 2nd and 3rd projects can use same code.

**Time limit**

- 5 days

# Submission status

Submission status	No attempt
Grading status	Not graded

Add submission

Make changes to your submission

**NAVIGATION**



- Home
  - My home
  - Site pages
  - My profile
  - Current course
    - jdi
    - Participants
    - Badges

Java Developer Induction

Java Starters

Good coding

Multithreading, Collections, Generics

 Multithreading

 Multithreading Quiz

 Collections, Generics

 Generics Quiz

 Collections Quiz

 Annotations Quiz

 MySQL

 MySQL Quiz

 PLSQL Quiz

 **Mini Project: Home Job Marketplace**

 UI

 HTML Quiz

 CSS Quiz

 Javascript Quiz

 Servelets

 Servelets Quiz

 JSP Quiz

 JSTL Quiz

 Mini Project: Home Job Marketplace - Phase 2

DB, UI, Frameworks

Tools

Performance

Care.com QA Induction Program

Topic 8

Topic 9

Topic 10

My courses

## ADMINISTRATION



Course administration

My profile settings

## BLOG MENU



View all entries about this Assignment

View my entries about this Assignment

Add an entry about this Assignment

View all of my entries

Add a new entry

Search

---

You are logged in as Swastik Padhi (Log out)  
jdi