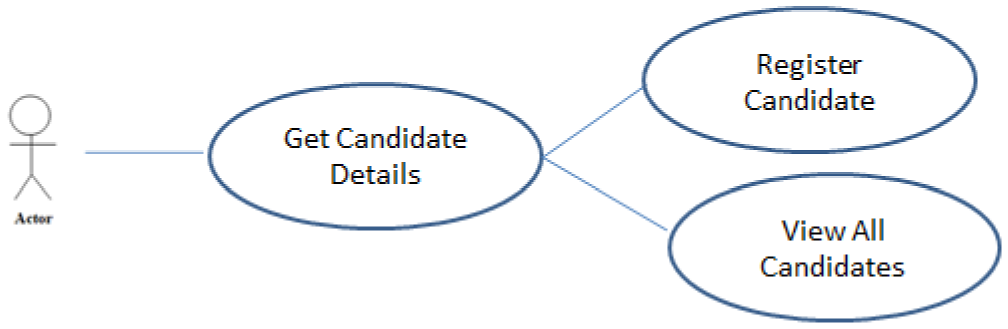
EXLNT Consultancy - Recruitment Management

1.0 Introduction

1.1 Functional Requirements

EXLNT Consultancy is well known for the recruitment services provided by them. The database of the recruitment details has increased drastically. So they need a web application to enhance their process. Create a Spring MVC Spring Boot Application to achieve this task. Design a Page to get candidate details. On clicking Register button, the application should add the candidate details into a list and maintain the details for further processing.

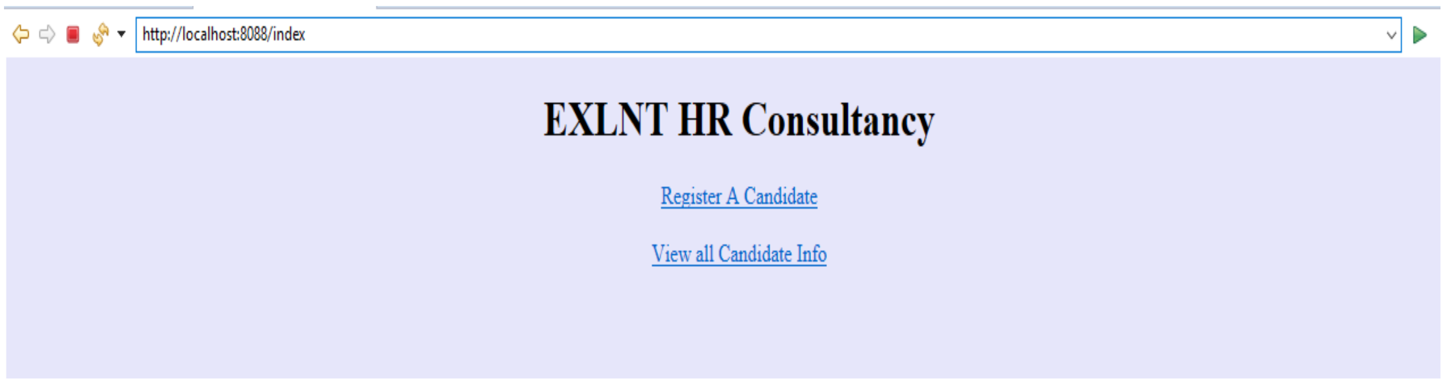
1.2. Use Case Diagram



2.0 Technical Specifications

2.1 Process Flow

- The first page of the application is **index.jsp** page, this page should have hyperlinks to navigate to **registerCandidate.jsp** and **viewAllCandidate.jsp**



EXLNT HR Consultancy

Register A Candidate

View all Candidate Info

- In **registerCandidate.jsp**, the page should have the below form components.

Component	Component Type	Component ID

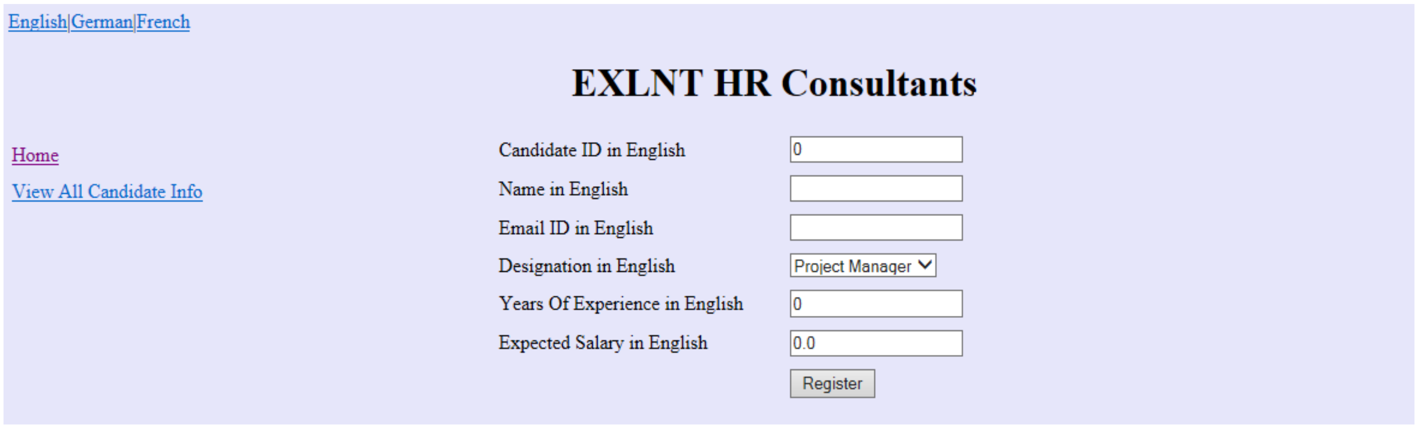
Candidate ID	Textbox	candidateId
Name	Textbox	name
Email ID	Textbox	emailId
Designation	Drop down	designation
Years of Experience	Textbox	yearsOfExperience
Expected Salary	Textbox	expectedSalary
Submit	Submit	submit

- In the same page we have the provision to choose language English, German and French. By default, the page content will be displayed in English. If we choose the language as French the label content and the error message in the page should displayed in French and if we choose the language as German the label content and the error message in the page should displayed in German. Ensure you implement the concept of Internationalization.
- In the **registerCandidate.jsp** page, ensure you use the component type and id correctly as specified in the above table.
- The values in the **designation** drop-down must be auto populated from the **controller** with the values as given in the below table. They should not be populated / hardcoded inside the JSP.

Designation
Project Manager
Team Lead
Developer
Tester
Administrator
Accountant

- **RecruitmentService** class contains the attribute **candidateList** with type as ArrayList<CandidateInfo>

- Sample **registerCandidate.jsp** for displaying the page content in English

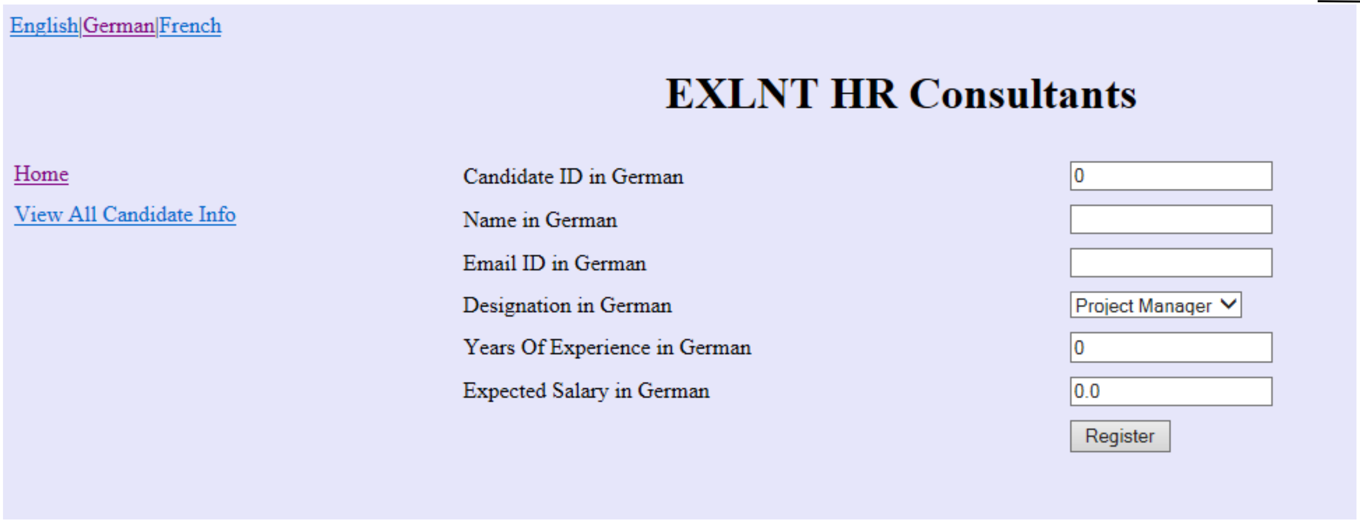

English|German|French

EXLNT HR Consultants

Home
View All Candidate Info

Candidate ID in English	0
Name in English	
Email ID in English	
Designation in English	Project Manager
Years Of Experience in English	0
Expected Salary in English	0.0
	Register

- Sample **registerCandidate.jsp** for displaying the page content in German

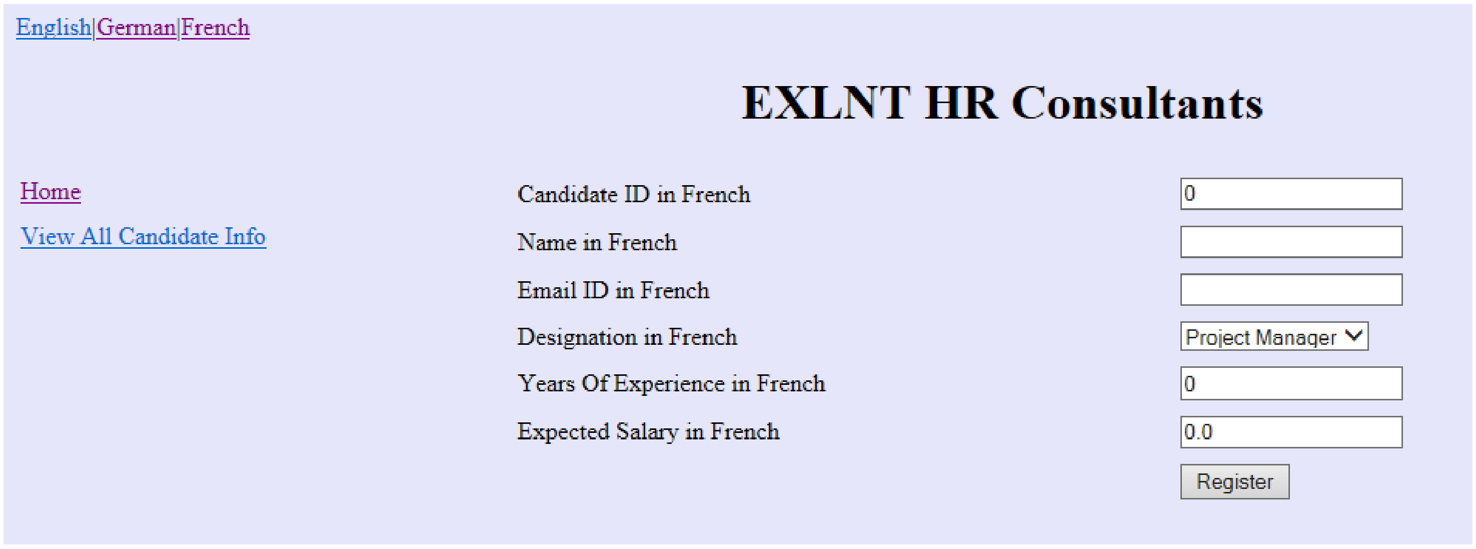

English|German|French

EXLNT HR Consultants

Home
View All Candidate Info

Candidate ID in German	0
Name in German	
Email ID in German	
Designation in German	Project Manager
Years Of Experience in German	0
Expected Salary in German	0.0
	Register

- Sample **registerCandidate.jsp** for displaying the page content in French


English|German|French

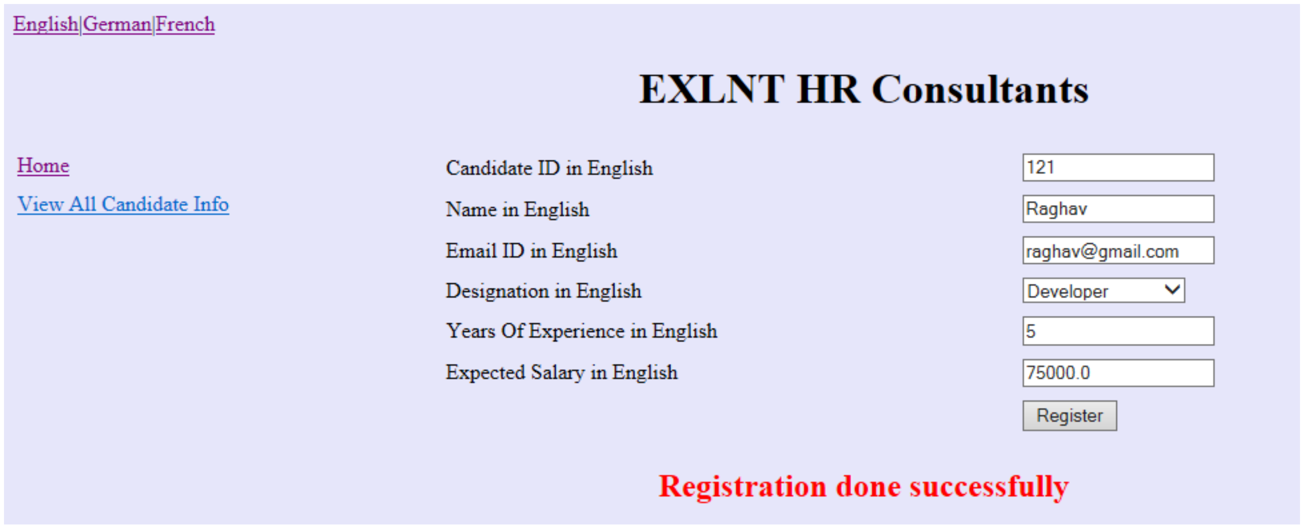
EXLNT HR Consultants

Home
View All Candidate Info

Candidate ID in French	0
Name in French	
Email ID in French	
Designation in French	Project Manager
Years Of Experience in French	0
Expected Salary in French	0.0
	Register

On-click of the Register button, after all successful validations (Refer 2.2 for validation), invoke **addCandidate** method of **RecruitmentService** class. Here iterate the candidateList and confirm whether it has a CandidateInfo object with same email Id. If not then add CandidateInfo object into the candidateList. Otherwise raise an **InvalidCandidateException** with the message **"Candidate <<name>> has already registered".** Handle this exception using **ExceptionHandler** and render the message in **exceptionPage.jsp**

registerCandidate.jsp


English|German|French

EXLNT HR Consultants

Home
View All Candidate Info

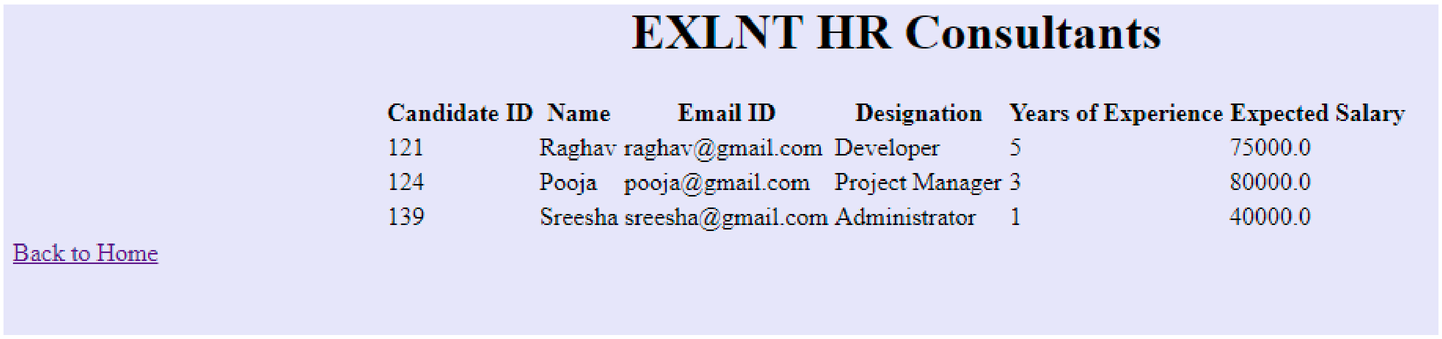
Candidate ID in English	121
Name in English	Raghav
Email ID in English	raghav@gmail.com
Designation in English	Developer
Years Of Experience in English	5
Expected Salary in English	75000.0

Register

Registration done successfully

After adding the CandidateInfo object into the candidateList, display the success message **"Registration done successfully"** in the div tag (refer the above screenshot). **div** tag with id as "**message**" is already available as part of code skeleton. While click on the hyper link H**ome**, it should navigate to **index.jsp**. And for **View All Candidate Info** hyper link , it should navigate to **viewAllCandidate.jsp.**

- Render the candidate list in the **viewAllCandidate.jsp**

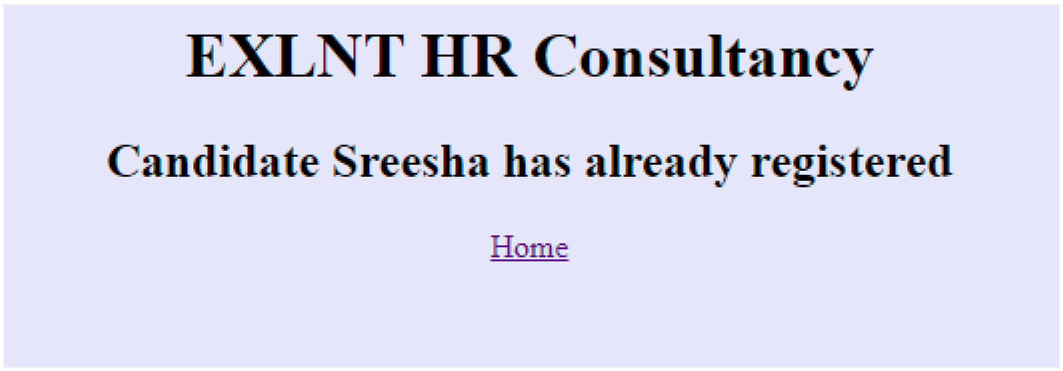

EXLNT HR Consultants

Candidate ID	Name	Email ID	Designation	Years of Experience	Expected Salary
121	Raghav	raghav@gmail.com	Developer	5	75000.0
124	Pooja	pooja@gmail.com	Project Manager	3	80000.0
139	Sreesha	sreesha@gmail.com	Administrator	1	40000.0

Back to Home

On clicking “Back to home” it should redirect to index.jsp page

- **exceptionPage.jsp** (while adding candidate details, if any duplicate email id found then render the message in exceptionPage.jsp)



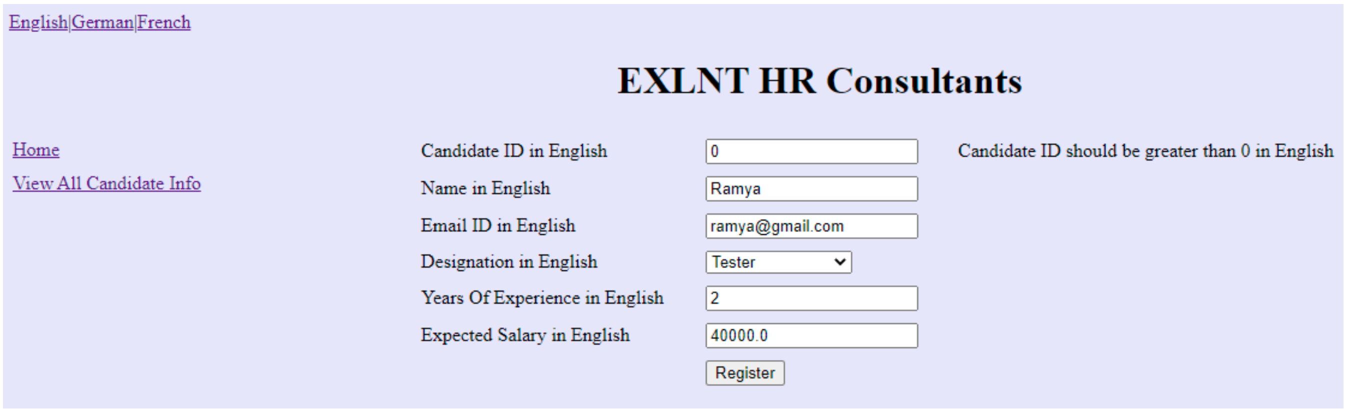
EXLNT HR Consultancy

Candidate Sreesha has already registered

On clicking “**Home**” it should redirect to index.jsp page

2.2Business Validation

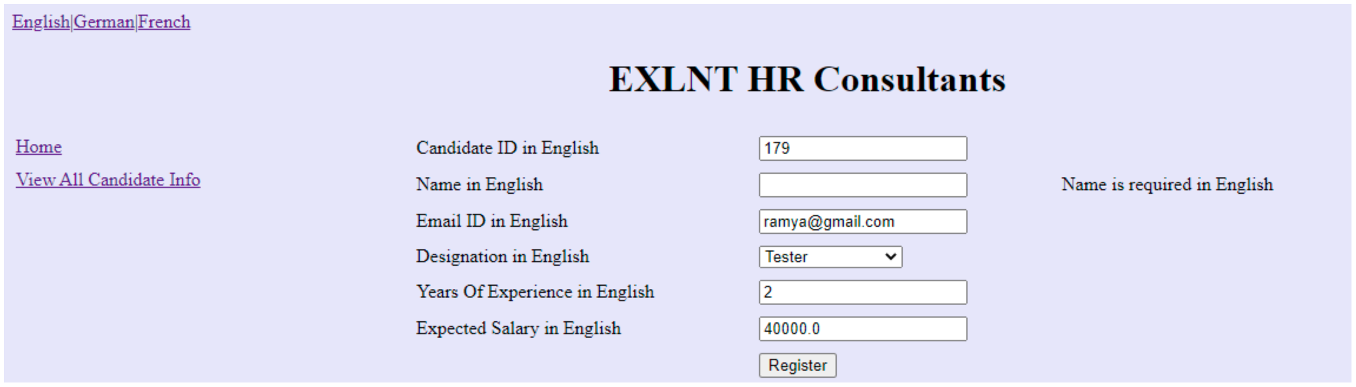
- **Rule:** Candidate ID should be a positive non-zero number
- **Message:** Candidate ID should be greater than 0 in <<locale>>
    - If validation fails, UI should display the error message - “Candidate ID should be greater than 0 in <<locale>>”.



EXLNT HR Consultants

Candidate ID in English	0	Candidate ID should be greater than 0 in English
Name in English	Ramya	
Email ID in English	ramya@gmail.com	
Designation in English	Tester	
Years Of Experience in English	2	
Expected Salary in English	40000.0	

- **Rule:** Name should not be blank
- **Message:** Name is required in <<locale>>
    - If validation fails, UI should display the error message - “Name is required in <<locale>>”.

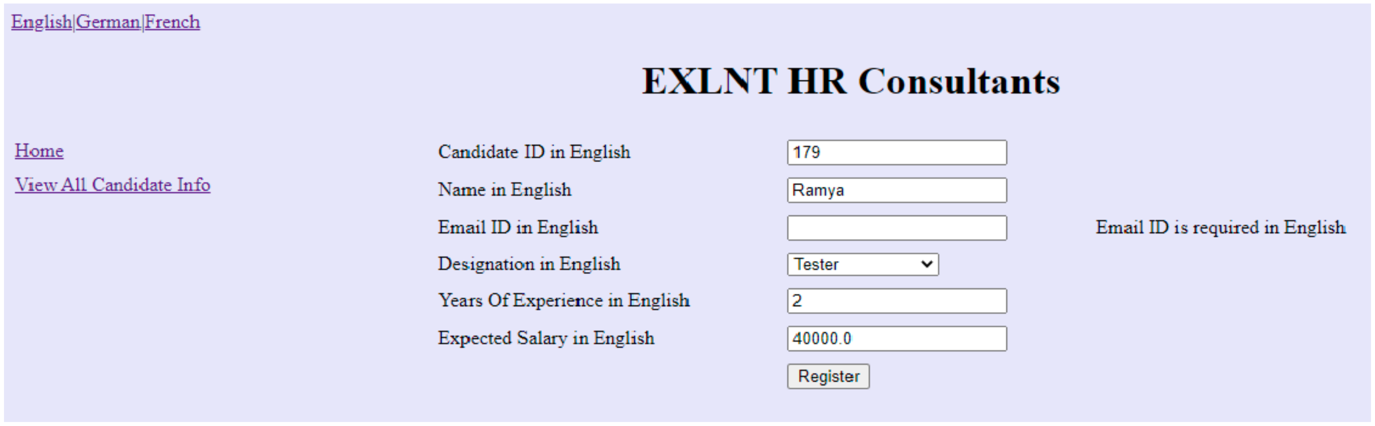

English|German|French

EXLNT HR Consultants

Home
View All Candidate Info

Candidate ID in English	179	
Name in English		Name is required in English
Email ID in English	ramya@gmail.com	
Designation in English	Tester	
Years Of Experience in English	2	
Expected Salary in English	40000.0	

- **Rule:** Email ID should not be blank
- **Message:** Email ID is required in <<locale>>
    - If validation fails, UI should display the error message - "Email ID is required in <<locale>>".

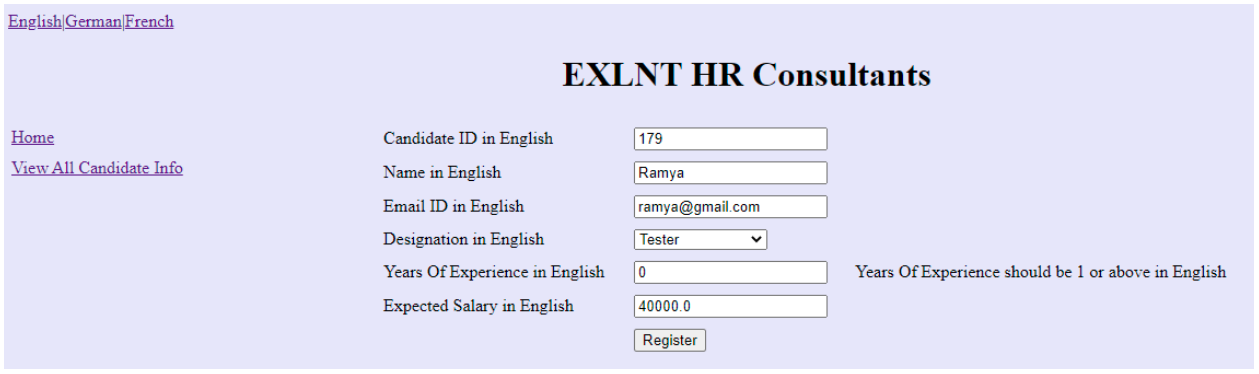

English|German|French

EXLNT HR Consultants

Home
View All Candidate Info

Candidate ID in English	179	
Name in English	Ramya	
Email ID in English		Email ID is required in English
Designation in English	Tester	
Years Of Experience in English	2	
Expected Salary in English	40000.0	

- **Rule:** Years Of Experience should be a positive non-zero number
- **Message:** Years Of Experience should be 1 or above in <<locale>>
    - If validation fails, UI should display the error message - "Years Of Experience should be 1 or above in <<locale>>".

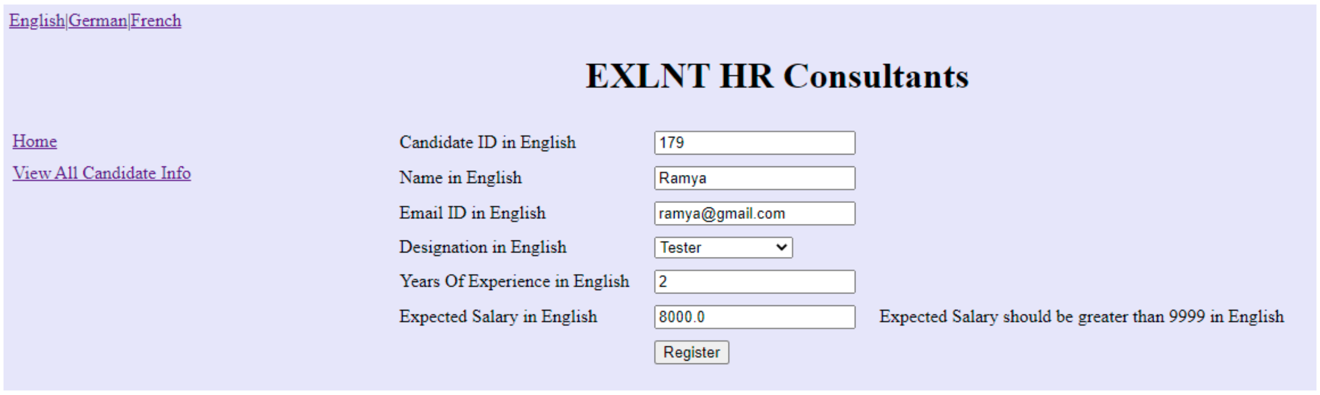

English|German|French

EXLNT HR Consultants

Home
View All Candidate Info

Candidate ID in English	179	
Name in English	Ramya	
Email ID in English	ramya@gmail.com	
Designation in English	Tester	
Years Of Experience in English	0	Years Of Experience should be 1 or above in English
Expected Salary in English	40000.0	

- Rule: Expected Salary should be greater than 9999
- Message: Expected Salary should be greater than 9999 in <<locale>>
    - If validation fails, UI should display the error message - "Expected Salary should be greater than 9999 in <<locale>>".


English|German|French

EXLNT HR Consultants

Home
View All Candidate Info

Candidate ID in English	179
Name in English	Ramya
Email ID in English	ramya@gmail.com
Designation in English	Tester
Years Of Experience in English	2
Expected Salary in English	8000.0

Expected Salary should be greater than 9999 in English

Register

- **Note:** The messages have to be retrieved from the appropriate properties file. The properties file has been given as part of code skeleton. Do not change/modify the property file

2.2Control Flow

- By giving the request **/index** from the browser (Eg: http://localhost:8088/index),the user should be redirected to the index.jsp
- On clicking Register A Candidate, it should navigate to registerCandidate.jsp page
- Get input for Candidate ID, Name, Email ID, Designation, Years Of Experience and Expected Salary
- On clicking **Register** button, do the validation for UI components
- Check the emailId in the candidateList, If there is a CandidateInfo object present with same email ID then raise **InvalidCandidateException**. Else add the CandidateInfo object into the list
- On clicking View all CandidateInfo, it should navigate to viewAllCandidate.jsp page
- Display the candidate list

3.0Component Specification

3.0.1Controller

- The RecruitmentController should be configured via annotation as a Controller

RecruitmentController			
Attribute Name	**Attribute Type**	**Access Specifier**	**Constraints**

recruitmentService	RecruitmentService	private	Use annotation to autowire

Method Name	**Method Argument name:type**	**Return type**	**RequestMapping URL & Request Method**
showIndexPage		String	/index& GET
showRegisterCandidatePage	modelAttribute "candidate": CandidateInfo	String	/showRegisterCandidate & GET
addCandidateToList	modelAttribute "candidate": CandidateInfo, result:BindingResult, model:ModelMap	String	/registerCandidate & POST
viewAllCandidateInfo	model:ModelMap	String	/viewAllCandidate & GET
populateDesignation		List<String>	Should be annotated with ModelAttribute with name "designation"
exceptionHandler	Exception	ModelAndView	Use annotation to handle the user defined Exception

- Inside the populateDesignation method you should add the designation into the List and return the List.
- **RecruitmentService** class should be autowired inside the Controller via annotations

3.0.2 Service

- **RecruitmentService** class should be configured via annotation as Service

RecruitmentService			
Attribute Name	**Attribute Type**	**Access Specifier**	**Constraints**
candidateList	ArrayList<CandidateInfo>	private	

Method Name	**Method Argument name:type**	**Return type**	**Responsibilities**

addCandidate	“candidate”: CandidateInfo	boolean	Check the emailId and raise InvalidCandidateException when a CandidateInfo object is present in candidateList with same emailId
viewAllCandidates		ArrayList<CandidateInfo>	Return the list of CandidateInfo in candidateList.

3.0.3Model

Class(Model)	Property	Methods
CandidateInfo	candidateId:String name:String emailId:String designation:String yearsOfExperience:int expectedSalary: double	//getters and setters are provided

- Include the needed annotations inside the CandidateInfo class for validations already mentioned. The error messages should be retrieved from the properties file

3.0.4 Internationalization

- Override the methods in InternationalizationConfig class provided as part of the code skeleton to support Internationalization.
- **Note**: For the LocaleChangeInterceptor, to internationalize the page based on the new Locale, use the parameter "language".
- **Hint**: Request from the UI (registerCandidate.jsp) for all the three hyperlinks have language parameter added to a request. Do not modify/change the hyper link /the values already provided as part of the code skeleton in the registerCandidate.jsp

3.1 Request Mapping Tasks

/index	Redirect the user to **index.jsp**
/showRegisterCandidate	Redirect the user to registerCandidate.jsp. Should invoke **addCandidateToList** method of the Controller after submitting

	all the valid Candidate details from **registerCandidate**.jsp, which in turn should invoke the **addCandidate** method of the RecruitmentService and if there are validation errors it should be redirected to **registerCandidate.jsp** page. If there is an exception then it should redirect to **exceptionPage.jsp.** If all input are valid it should be redirected to registerCandidate.jsp with appropriate message.
/ viewAllCandidate	Should invoke **viewAllCandidateInfo method** of the Controller which in turn should invoke the viewAllCandidates of the RecruitmentService .

4.0 Overall Design Constraints

1) Implement the code using the best design standards.
2) Use Internationalization for all the labels and messages in the Register Candidate page.
3) registerCandidate should support Internationalization for three languages English, German and French. The error messages and the labels should be displayed from the properties file depending on the locale. **The properties file is already provided as part of the code skeleton.**
4) Spring Validator should be used for all the Validations. Use only annotation for performing the validations.
5) RecruitmentManagementSystemApplication which is the main class, is already provided as part of the code skeleton. Include only the required annotations to auto scan the Controller, Service classes.
6) **Do not change the property name given in the application.properties files, you can change the value and you can include additional property if needed.**
7) **In the pom.xml you are provided with all the dependencies needed for developing the application.**
8) You will not be evaluated based on the UI design (layout, color, formatting, etc.). You are free to have a basic UI with all the required UI components (input fields, buttons, labels, etc.). Expected components with the id alone should be designed as per the requirement.
9) Adhere to the design specifications mentioned in the case study.
10) Do not change or delete the class/method names or return types which are provided to you as a part of the base code skeleton.
11) Also do not change the name or id of the HTML component which are provided to you as a part of the base code skeleton.
12) **Please make sure that your code does not have any compilation errors while submitting your case study solution.**