Feefo Software Engineering Technical Assessment

Here at Feefo, we want to give candidate software engineers the opportunity to present the best of their skills and knowledge.

With this in mind, we have 3 part technical assessment which we ask candidates to complete to support their application. We would prefer if a candidate completed all 3 assessments, but appreciate an assessment may be in an unfamiliar domain; for example, you may be thinking "*I don't know React yet!"* In the case an assessment is not completed please endeavor to ***shine*** in those you can complete, and we just ask for a simple rationale of the reason instead.

Candidates successfully passing the assessment stage will be invited for an interview, please review your submission prior to the interview so that you are ready to answer technical questions.

1. Object Orientated Programming

Please complete the problems in Java or another Object Orientated language.

Use your favorite IDE to create and test your solutions, including any testing descriptions/evidence.

Please provide your solution as a runnable and testable project, either as a public GitHub repo or zipped project files, so that they can be imported and tested in an IDE.

We are looking for a good project structure, evidence of good Object Orientated Programming Principles, defensive coding, and unit testing.

Normalizing Job Titles process

Provided with a list of ideal (normalized) job titles, create a class that implements a process that returns the best match when provided with an input string.

Concretely, given a normalized job titles list of "Architect", "Software engineer", "Quantity surveyor", and "Accountant", write a process that returns the normalized result for the input.

input		**normalized**
"Java engineer"	>	"Software engineer"
"C# engineer"	>	"Software engineer"
"Accountant"	>	"Accountant"
"Chief Accountant"	>	"Accountant"

Hint: internally in the process, consider a quality score q, where 0.0 <= q <= 1.0, to find the closest match.

Here is some sample code that would call your code:

```
String jt = "Java engineer";
Normaliser n = new Normaliser();
String normalisedTitle = n.normalise(jt);
//output normalisedTitle
jt = "C# engineer";
normalisedTitle = n.normalise(jt);
//output normalisedTitle

jt = "Chief Accountant";
normalisedTitle = n.normalise(jt);
//output normalisedTitle
```

2. UI Assessment

Please see the the attachment `ui-assessment-master.zip`

3. Web App Restful API System Design

We operate small cross-functional scrum teams, as such it's important to us that you can understand requirements and design solutions to business problems.

From this exercise we are looking to understand how well you:

- Understand the problem space
- Understand any technical limitations
- Identify any assumptions
- Design a solution
- Present that solution
- Expand upon that idea in discussion

Please provide responses for all the numbered sections. Please submit your design in a single document, or as a GitHub repo wiki.

In the interview, we will allow time for you to present your system design, and take questions.

A note web app

Consider a simple note web app, where a user can write a note, save a note, view a list of their notes, and delete a note. The user's notes are saved so that they are available via any web-capable device.

We are not asking for a prototype solution, descriptions, diagrams and pseudo code would be ideal.

Acceptance Criteria

- the ***note app*** runs in a client browser
- **User** enters a block of text
    - Save button saves the note
- **User** can see a list of saved notes
- **User** can delete a saved note.

Out of scope

- User ***note app*** account registration
- User ***note app*** account log in

Assumptions

- The user must be logged in to use the ***note app***
    - login is required for the user to view their saved notes
    - login is required for the user to save a new note
- The web app maintains a user session

1. Describe high level design

Show the main ***note app*** components and the logical interactions that will fulfill the requirements.

2. Web App UI

Provide a wireframe design of the note web app that will fulfill the requirements.

- Consider what UI components are required and how these interact with the other components.
- What (if any) validation is required?

3. Data Model

Describe how a note will be modelled

- consider the required properties

4. Restful API

Describe the Restful API required to fulfill the note app.

- how would the web app get the user's notes?
- how would the web app save a user note?
- what are the URL for the note resource(s)?
- and verbs to expose the actions?

5. Web Server

Describe how the webserver implements that Restful API:

- consider how each action will be implemented
- what (if any) business logic is required?
- how are the notes saved?