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 technical assessments 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 a 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.

1. Algorithms

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.

Ideally provide your solutions as a public GitHub repo (or project(s) files) so that it can be imported and tested in an IDE.

Problem 1

Provided with a list of ideal (normalized) job titles, implement a process which returns the best match when provided with an un-normalized input string.

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

un-normalized		**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
```

Problem 2

Provided with an array of integers of variable length (int[] input), implement a process that calculates the (1) median, (2) mean, (3) mode and (4) range for the array and outputs these four values to the command line.

Here is some sample code that would call your code:

```
//load the results from a big computation into mybigarray
int[] mybigarray = generateSomeHugeArray();

//now print out some basic stats on the array
YourClass arrayManipulator = new YourClass();
arrayManipulator.getStats(mybigarray);
```

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:

- Research 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 submitted your design in a single document, or as a GitHub repo wiki.

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 web server implements that Restful API

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

6. Further questions to consider

- what user login and session strategies could be used?
- how would your UI solution facilitate extension and reuse?
- what as considered when deciding on a storage component?
- how might the capability for a user to share a note publicly affect the design?
- what additional capabilities would be useful, and how would these affect the design?