



# Senior Software Engineer

Coding Challenge - Backend

Version Date: May 2024

# Senior Software Engineer

## Coding Challenge and Review – Backend

Congratulations! We would like to invite you to complete our Backend Software Engineer coding challenge.

In this stage you will prepare a solution to our coding challenge for us to review with you during a follow-up interview.

If the solution is suitable and meets the requirements you will be invited to have a review session to deep dive into your solution and discuss an extension problem. Please read the attached materials (slides 3-4) in the presentation deck to help you prepare for this interview.

# Things to Prepare

## Your Code Challenge Solution

You can choose any language, provided it showcases your OOP and code design skills - TypeScript, Ruby, PHP are preferred given their use at Shippit. Avoid pure scripting languages like Perl or Shell.

We recommend you implement your solution in the following steps:

- Data modeling
- Solution and test for AddChild
- Solution and test for GetRelationship

Submit code for review and execution to [chris.duffy@shippit.com](mailto:chris.duffy@shippit.com)

## Review Session

Have your code solution ready to share and talk through during the follow-up session. In this session an extension problem will be presented for you to discuss with our engineers.

We want you to be comfortable during the interview! Feel free to use your favourite IDE to help you code and raise any questions with us early! Remember Development is a collaborative effort in all phases.

# What do we assess during the interview?

## Thought Process

Make sure you let us know your thoughts, trying to think out loud. That way we can also help you look in the right direction. Take time to assess the problem and talk us through the solution.

## Interaction

We love team players, especially those who listen to feedback and can work with the team to achieve a better outcome through a bias to learning and collaboration.

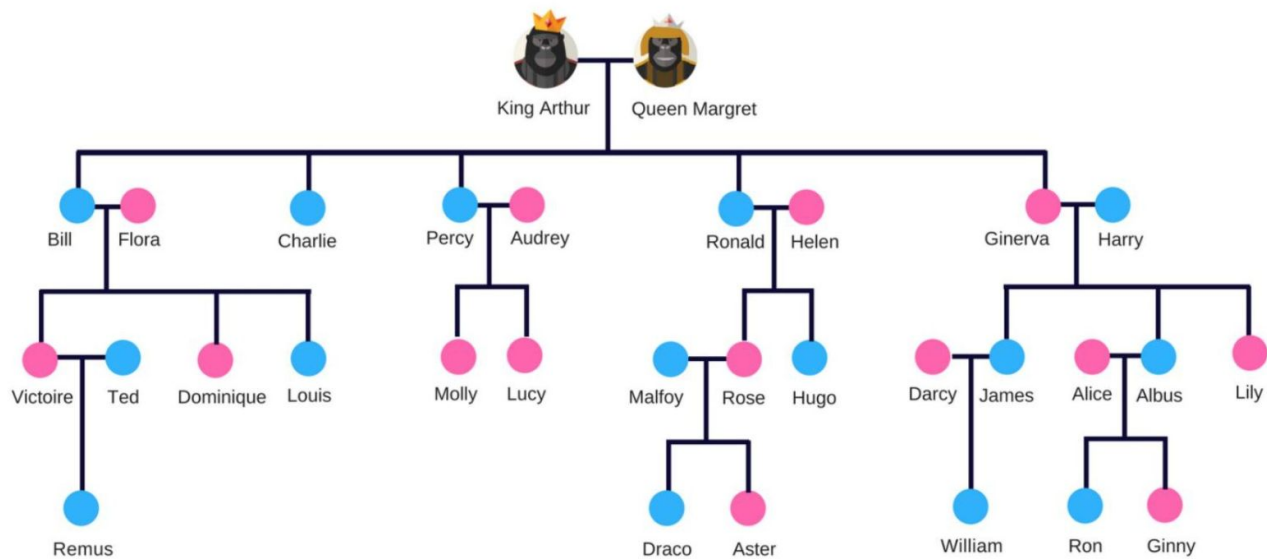
## Tech best practices

Ability to demonstrate some of the best practices, such as:

- OOP and/or functional design patterns
- Effective use of common design patterns
- TDD and quality prioritises
- SOLID principles
- A want to understand the real world value of software

# Tech Test Challenge

## Family Tree



The Arthur Family Tree

● Male ● Female

# Tech Test Challenge

Our problem is set on the planet of Lengaburu, the distant galaxy of Tara B. Our protagonists are King Arthur, Queen Margaret and their family.

King Arthur is the emperor of Lengaburu and has been ruling the planet for the last 350 years (they have long lives in Lengaburu!). Let's write some code to get to know the family.

The coding problem is for backend and full stack engineers as it tests the object oriented fundamentals.

# Tech Test Challenge



## Meet the Family

The solution should model out the King Arthur family tree so that:

- You should be able to add a child to any family in the tree through the mother
- Given a 'name' and a 'relationship', you should output the people corresponding to the relationship in the order in which they are added to the family tree

Simple right?

Remember, our evaluation is based not only on getting the right output, but on how you come up with the solution.

# Tech Test Challenge



## Sample Input/Output

Input needs to be read from a text file. and output should be printed to console.  
Your program should take the location to the test file as a parameter.

↻ Input format to add a child:

```
ADD_CHILD "Mother's-Name" "Child's-Name" "Gender"
```

↻ Input format to find the people belonging to a relationship:

```
GET_RELATIONSHIP "Name" "Relationship"
```

↻ Output format on finding the relationship:

```
"Name 1" "Name 2"... "Name N"
```

↻ Example test file:

```
ADD_CHILD Flora Minerva Female  
GET_RELATIONSHIP Remus Maternal-Aunt  
GET_RELATIONSHIP Minerva Siblings
```

↻ Output on finding the relationship:

```
CHILD_ADDED  
Dominique Minerva  
Victoire Dominique Louis
```



# Tech Test Challenge



## Sample Input/Output

More sample output scenarios

More sample output scenarios.

### ↻ Sample 1

```
ADD_CHILD  Luna  Lola Female
GET_RELATIONSHIP  Luna Maternal-Aunt
```

### ↻ Sample 2

```
ADD_CHILD  Ted  Bella Female
GET_RELATIONSHIP  Remus Siblings
```

### ↻ Sample 3

```
GET_RELATIONSHIP  Lily Sister-In-Law
```

### ↻ Sample 1 Output:

```
PERSON_NOT_FOUND
PERSON_NOT_FOUND
```

Luna does not exist in the family tree

### ↻ Sample 2 Output:

```
CHILD_ADDITION_FAILED
NONE
```

Ted is male, hence child addition failed

### ↻ Sample 3 Output:

```
Darcy Alice
```

# Tech Test Challenge



## Relationships to Handle

There are many relationships that could exist and here are some of the relationships you may want to implement.

You need to ensure your design is extensible enough to add more relationships in the future

Relationships	Paternal-Uncle	Maternal-Uncle	Paternal-Aunt	Maternal-Aunt	Sister-In-Law	Brother-In-Law	Son	Daughter	Siblings
Definition	Father's brothers	Mother's brothers	Father's sisters	Mother's sisters	Spouse's sisters, Wives of siblings	Spouse's brothers, Husbands of siblings			



**Looking forward to  
meeting you!  
Feel free to reach  
out if you have any  
questions**