

# Introduction

---

Hello Dear Candidate,

Firstly, we would like to drop several lines about us and our project. We are a team of experts who started an enterprise level solution from scratch. It means, all technology and management decisions are taken on our side. But, as you know, nothing is for free. That mean we have great responsibility for decisions that we made and for the people who would like to join us.

We learn quickly, we have a passion for what we do and how we do it. Besides, we are looking for people with the same culture and values like we have. We are growing, we learn new technologies and approaches. About 8 people (25% of the whole dev team) in our team were promoted to higher position (from junior to middle, from middle to senior). We believe, that it is the perfect place to start.

Before you start to think about how to do the technical task, you need

- 1) Do you know how to write testable code?
- 2) Are you able to learn quickly?
- 3) Do you know how to write maintainable code?
- 4) Are you smart?
- 5) Do you see weak points in your solution?
- 6) Are you able to get things done?
- 7) Do you not fear complex tasks (big, complex, and initially you have no idea how to solve it)?

If you have at least 6/7 yes, let's start=) If no, just skip it and leave competition.

Best regards,  
CSC Team.

# Business requirements

---

## Junior/Intern level:

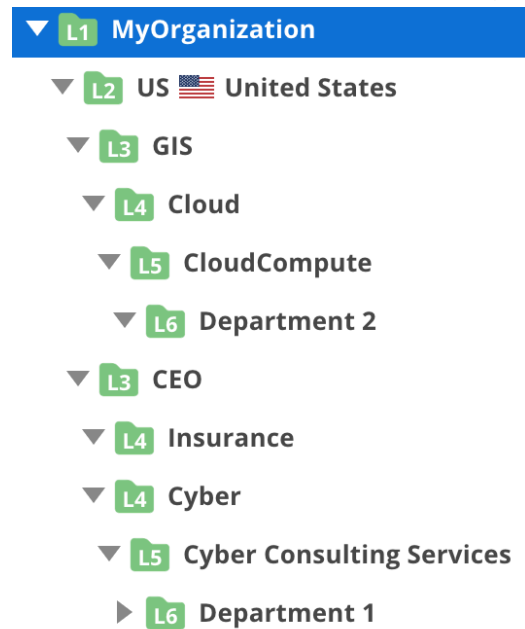
As a user, I would like to create a hierarchical tree structure to organize my assets in my enterprise. So, I will have following entities inside the system:

1. User (Name, Surname, Email, Address)
2. Country (Name, Code)
3. Organization (Name, Code, Organization Type (General Partnership, Limited partnerships, Limited Liability Company (Co. Ltd.), Incorporated company, Social enterprise, Other), Owner).
4. Business (Name)
5. Family (depends on business)
6. Offering (depends on family)
7. Department (name)

## Example of data for Business Family and Offering:

- Business: GIS -> Family: Data center -> Offering: Data storage/
- Business: GIS -> Family: Data center -> Offering: Data management
- Business: GIS -> Family: Cloud -> Offering: Biz cloud
- Business: GIS -> Family: Cloud -> Offering: Cloud compute
- Business: CEO -> Family: Cyber -> Offering: Consulting services

This structure needs to be displayed on the screen:



As a user, I would like:

- 1 See the tree.
- 2 Use expand/collapse all button  
**TIP:** Think about this requirement, what is hidden complexity here?
- 3 Create/update/delete an organization
- 4 Create/update/delete a country under organization. (**Validation rule:** Country must be unique inside an organization)
- 5 Create/update/delete a business inside country (**Validation rule:** the business has to be unique inside a country).
- 6 Create/update/delete a family inside business (**Validation rule:** the family must be unique inside a business)
- 7 Create/update/delete an offering inside family (**Validation rule:** the offering must be unique inside a family).
- 8 Create/update/delete a department (**Validation rule:** the department name must be unique inside the offering).
- 9 To see logs or exceptions that happened inside the system.
- 10 If a user delete upper level, all bottom levels needs to be deleted.
- 11 User must be authenticated using external providers (chose one Google/LinkedIn/Facebook/Twitter),
- 12 All GET methods must be accessible without authentication.

### Middle level (Optional):

Complex requirement (you need to choose one option and implement it, and consider how to implement other):

- Pivot tree (you can check how pivoting works in excel) by levels (As a user I would like to see the same tree but starting from level 5)

**TIP:** Check how excel works with pivot tables.

- Filter the tree using business, family, offering, country, department and organization type. Assume that user is free to choose what properties is needed to filter. (For example: If user filtered out the country, all bottom levels will disappear from the hierarchy).

### Nonfunctional requirements:

---

1. Code coverage must be more than 70%.
2. **No UI is required**, only API. (solution must be without UI).
3. API must work with JSON
4. You need to provide the description of your API using one of suggested solutions: [Swagger](#)/[Blueprint](#). Please include all JSON templates for POST, PUT routes with instruction how to test your API (instruction must be placed into [\\*.md file](#)).
5. All code must be submitted to the public [github](#) repository.
6. You will need to use following technology stack:
  - ASP .Net Core (Web API)
  - ORM: Entity Framework Core
  - IoC: Simple Injector.
  - Unit tests: XUnit, NSubstitute.

Be ready to answer on following questions:

- What are the potential bottlenecks of your solution?

- How can we measure and optimize performance of the application?
- How can be filtering implemented? What are pros and cons of the solution? **(optional)**

P.S., In any case, thanks in advance for your efforts, you will do to accomplish the current technical task. We are sure, if you finish this task, you will get new skills and knowledge, that can help to develop software products.

Good luck, and hope to see you on the interview.