RED ABACUS DEVELOPMENT SRL
Str. Gruia, nr. 27 400171
Cluj-Napoca, Romania
+40 748 077 177
www.abac.software

Coding challenge

Requirement

Thanks to the efforts put forth by brave explorers, we live in an era where the human race will become a multi-planet species. One of the challenges for doing this is keeping track of the status of each expedition that traverses multiple unknown solar systems in order to find life.

We plan to create an application that models the expedition process so that we can keep track of the planets that were visited, and the status of each of these planets.

Each **planet** in the solar system is visited by a **team of explorers** composed of **one human captain** and **multiple robots** that board a **shuttle** and navigate the solar system for finding planets that can sustain life. When reaching a planet, the robots can determine whether it is suitable for human life or not. The human captain can use the application in order to communicate the status of the expedition to the other exploring teams, so that they do not visit the same planet again.

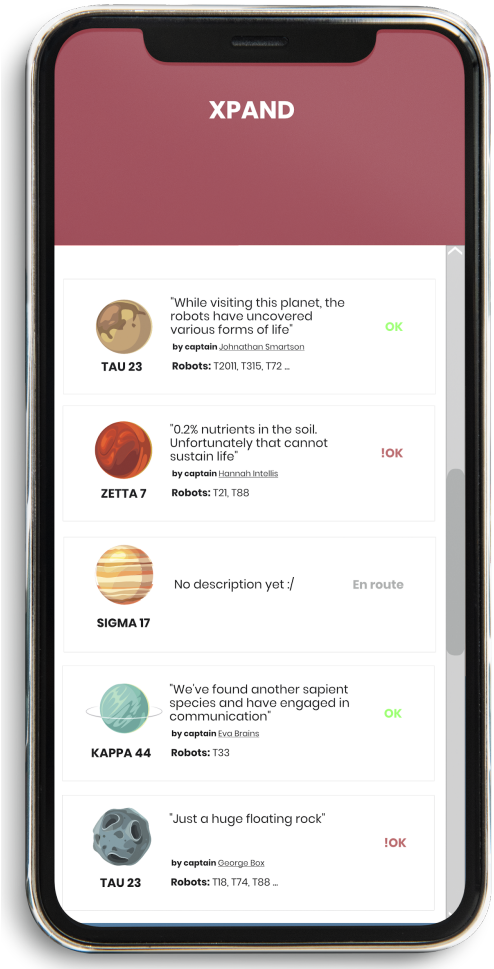
IMPORTANT: Please focus on project structure and not on completion, as we are interested in your modeling and problem solving abilities.

IMPORTANT: Any conceptual changes to the way the modelling is performed are more than welcome.

RED ABACUS DEVELOPMENT SRL
Str. Gruia, nr. 27 400171
Cluj-Napoca, Romania
+40 748 077 177
www.abac.software

The UI

The user interface for the application should be a web based, single page application that runs on phone resolution, written in whatever is most convenient (**Angular**, **React**, **Vue** or without any frameworks, and should look something like:



The application practically allows the user to do two things:

1. Allows the user to see all the planets existent in the solar system, in order to see their statuses (UX close to what is shown in the mockups)
2. Allows the captain to change the status and description of a planet (this is up to you to figure out how exactly)

A planet is represented in the system by the following aspects:

- The planet has a name
- The planet has an image
- The planet has a description that is filled out by the captain of the team
- The planet has a status: OK, !OK, TODO, En route
- The planet has a number of robots that are exploring it

The tasks that are part of this UI challenge are:

- Start an application from scratch in a chosen technology
- Implement a way of retrieving the **planet** data from the API
- Implement the UI screen as shown in the mockup
- Implement a way in which the user of the app can update the properties of the planet

The API

The backend system (or API) should be a web service written in **.NET** that allows the consumers (UI application) to perform operations on **planets**.

The protocol for allowing the UI to perform the required operations can be either **REST** or any **websocket** driven implementation.

For persistence, any solution (SQL or NoSQL) is acceptable (some suggestions here are **SQLServer**, **MySQL** ,**Redis** etc)

At a conceptual level we know that:

- We have planets (:O)
- Planets can have statuses and descriptions added by captains of teams
- Teams contain a human captain and multiple robots

The tasks that are part of this API challenge are:

- Start an application from scratch in the mentioned technology
- Design de model for the mentioned problem
- Choose a storage solution that can handle persistence of the model
- Implement a means to expose the model to the UI using a chosen protocol