C# recruitment test for LongoMatch

What is this test about?

If you are reading this document then you're in the process of becoming a new member of our development team. Like any other company, at LongoMatch we are looking for the best engineers to join our team, but it's always hard to do so just by reading a CV or in a 1-hour interview, so like Linus Torvald said once: "Talk is cheap. Show me the code."

The goal of this test is to quickly understand how you would fit in our team by creating a very simple "ToDo list" application, using the same development stack and tools we use on a daily basis. As you will see, the test itself is quite simple and depending on your skills you should finish it in around 2-4 hours. It will try to cover some important aspects to us:

- Creation of a new project from scratch
- Use of git
- Use of async APIs
- Use of the design pattern MVVM/MVC/MVP/etc.
- Understanding of User Stories
- Whatever other thing you want to show us!

The rules

We will provide you with a description of the app, as it would be done by our Product Manager, and a list of User Stories describing its functionality. You will create the new application and publish your work in a public repository using a service like github or bitbucket. As you will see, the User Stories do not have any UI design, so design it as you want. We will value the ability to work with new API's, your autonomy to interpret and implement User Stories, the use of good design patterns and your ability to split your work into commits or branches.

The LongoToDo app

At LongoMatch we have decided to launch a new app that helps users keep a balanced life organizing themselves with a list of "ToDo" tasks called: LongoToDo. In a first iteration, we will release the application for the Android operating system, targeting API 29 as the minimum version, but we have plans in the future to release the same app for iOS too, so we have decided to use Xamarin/MAUI or Avalonia (up to you) as UI toolkit to have a high ratio of code reuse across platforms.

The LongoToDo app will allow users to perform the following tasks:

- List all the items of the ToDo list.
- Mark existing items as “Done”.
- Create new ToDo items.
- Delete existing items.

This application will use a service in the cloud to work with the ToDo items through a Rest API. The service is not live yet, but we have created a demo one that provides the same API as the final one so we can start developing and testing the application before we launch it. The code is stored in our github repo: https://github.com/fluendo/ToDoAPI
Running the application will start a new service at http://localhost:8080/api/todo exposing the ToDo API.

User stories

As a user, I want to list the ToDo items.

The application will start with a page view that lists all the ToDo items provided by the ToDo server API.

A ToDo item has 2 properties:

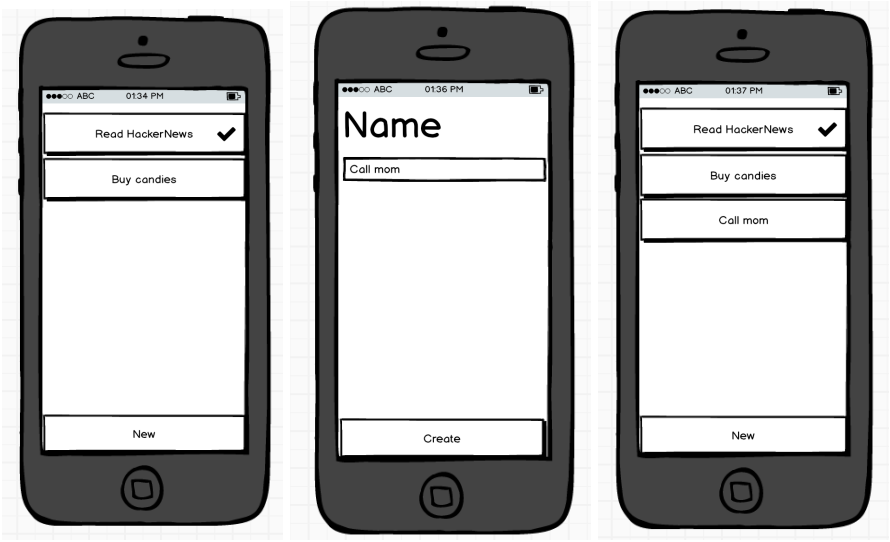
- Name: the name of the item
- IsComplete: true if the item is completed, false otherwise.

The ToDo items will be displayed in a list, one row per item with the name of the item and an icon that is displayed with the item is completed. The items will be listed in the same order provided by the API.

As a user, I want to create a new ToDo item.

Once we can list ToDo items in the application, the user should now be able to create new items.

A new button “New” will be added to the UI. When the user clicks this button, the application will navigate to a new page where the user can edit the name of the ToDo item. If the user clicks back in the navigation page, no items will be created, instead if the user clicks the button “Create” a new ToDo item will be added to the list. Newly created items start with the state “not completed”.



As a user, I want to delete a ToDo item.

A user should now be able to delete ToDo items using a context action for the cell with the name “Delete”. When the user clicks on the Delete button, the ToDo item will be removed and a new popup message will be presented to the user with the text “ToDo item {Name} has been deleted correctly”, replace {Name} with the name of the item.

As a user, I want to change the completion state of a ToDo item.

To keep track of the state of the ToDo items, the user should be able to toggle their state, from “not completed” to “completed” and vice-versa. The user can toggle the state of the ToDo item by tapping a cell in the list. If the item is completed an icon with the ✓ image is displayed at the end of the row.

As a user, I want to refresh the list of ToDo items.

Using the built-in pull-to-refresh functionality in Xamarin.Forms’ ListView, the user should be able to refresh the list of ToDo items from the service. To test it, you can launch a new instance of the app attacking the same service, creating a new item and refreshing the list in the other device.