

# iOS Test Assignment

## Task

Write an application that displays a list of employees from a remote server along with their details. If an employee exists in the phone contacts database, enable the user to open the native contacts display form as well.

### Mandatory application requirements:

The application has two screens. The first screen is a list view of all employees grouped by position. Groups are sorted alphabetically, employees are sorted by last name. Each employee is unique by their full name (first name + last name) and is displayed only once in the list view.

The user can use pull-to-refresh to update the data from the servers. If any errors occur with fetching or parsing the data, the list should display a generic error message (list should remain refreshable).

The application should fetch the phone's built-in contacts and match them against the list of employees. If the employee has a matching contact in the phone (first and last names are a case-insensitive match), the list element should contain a button which displays the iOS native Contact detail view. If no match is found the button should not be visible.

### Detail view:

Selecting a list element displays a detailed view of the employee's information. This includes their full name, email, phone number, position and a list of projects they have worked on.

If the employee matches a valid contact in the phone, a button is visible at the bottom of the screen which takes the user to the native Contact detail view. If no match is found the button should not be visible.

### Non-mandatory Application requirements

Save the last successful server response and display it on the next application launch. Network requests should still be executed whether the data is cached or not.

Implement search function. Search results should be displayed on matching first name, last name, email, project or position.

Write [README.md](#) file where you explain:

- What are the requirements to compile and run program

- What architecture pattern did you use (including reference) and why
- In case of complex solution/algorithms for some feature, please explain how it works

Write automated tests or provide a written plan on how a QA person test the app, pointing to the most critical functionality to be tested.

### Data specification:

The employee listings are available at [https://tallinn-jobapp.aw.ee/employee\\_list](https://tallinn-jobapp.aw.ee/employee_list) and [https://tartu-jobapp.aw.ee/employee\\_list](https://tartu-jobapp.aw.ee/employee_list). Both listings should be used.

Values are encoded in JSON with the following schema:

```
{
  employees: [
    {
      "fname": Peeter,
      "lname": Termomeeter,
      "contact_details": {
        "email": "peeter@telisa.ee",
        "phone": "55 555 555" (optional),
      },
      "position": IOS|ANDROID|WEB|PM|TESTER|SALES|OTHER,
      "projects": ["MyCoolApp", "OneTimeThing"] (optional)
    },
    {
      . . .
    }
  ]
}
```

### Technical requirements:

- You may use the latest stable version of Xcode or the latest developer beta.
- App must support the latest released version of iOS or the latest public beta.
- App must be built using the UIKit framework.
- You may not use third-party external dependencies.
- App must work on all iPhone screen sizes and in both screen orientations.