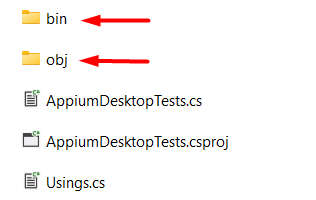
# QA Front-End Automation – "Contact Book" Exam



**Exam** assignment for the ["**QA Front-End Automation" Course @ SoftUn**i](https://softuni.bg/trainings/4022/qa-automation-front-end-march-2023).

You are given a zip archive, that contains the skeleton for your C# tasks. When you unzip it, open the **"ContactBookExam"** folder and open the **"ContactBookExam.sln"** file. This will open the ContactBookExam solution in your Visual Studio. It contains three projects: **AppiumDesktopTests, AppiumMobileTests** and **SeleniumTests**.   
**You will need to install the needed Nuget packages,** write your exam tests in the appropriate project and submit your work as a single zip / rar / 7z archive.   
**NB!** **Projects without Nuget Packages installed will be disqualified and automatically evaluated with 0 points.**

Keep in mind that you have to delete .bin and .obj folders for each project.



## The "Contact Book" System

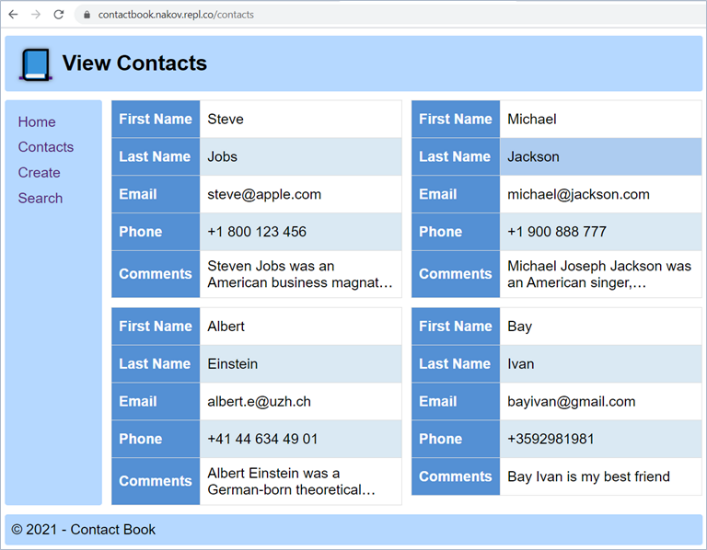
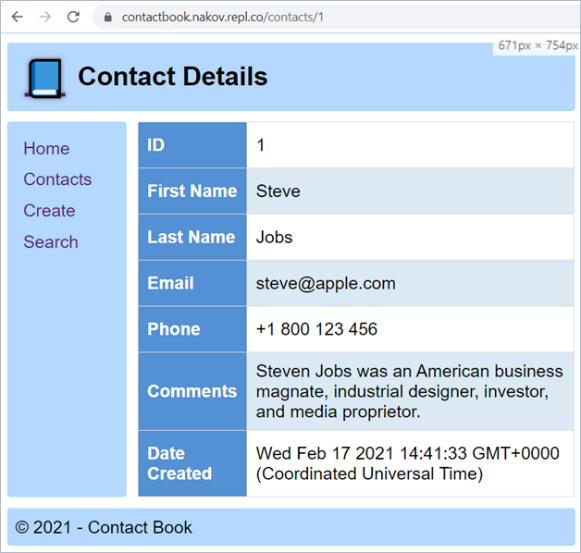
The "**Contact Book**" is a simple information system for managing **contacts**. Each contact consists of first name, last name, email, phone and comments. Users can **view** the contacts, **search** by keyword, and **add** new contacts.

You are given **Web** app + **Desktop** app + **Android** mobile app client for the task board system. Your assignment is to write **UI automated tests** for the system.

## Web App Functionality

The **"Contact Book" Web app** supports the following operations:

* Add a new contact by given contact data
* View all contacts
* View contact details
* Search contacts by keyword

## Installing and Running the App

You can check the application functionality on: [**https://contactbook.nakov.repl.co**](https://contactbook.nakov.repl.co)

To avoid conflicts, it is highly recommended that you **fork the project** for this app from:

**<https://repl.it/@nakov/contactbook>** into your **own repl.it account and run it from there.**

/**How to run the app on Repl.it** - ref. Running-The-Exam-App-On-Repl-it.docx/

Alternatively, you can **install** and **run** the app on your **local machine**:

|  |
| --- |
| git clone https://github.com/nakov/ContactBook  cd ContactBook  npm install  npm start |

## Resetting the App

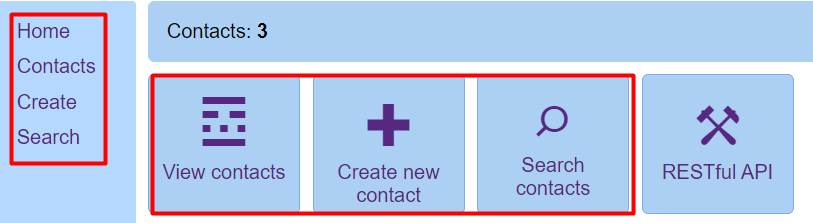
The app **does not have a persistent database** storage, so you can **reset it** by a simple **restart** (stop & start).

* After restart, you will lose all changes and the default sample data will be populated automatically.

## 1. Contact Book Web App: Automated Selenium UI Tests

Write **Selenium-based automated UI tests** for the "**Contact Book**" app. You should implement the following **automated UI tests**:

**For each test you will need to start from the Home page and navigate to the target page via side navigation bar link or by clicking on button icon link.**

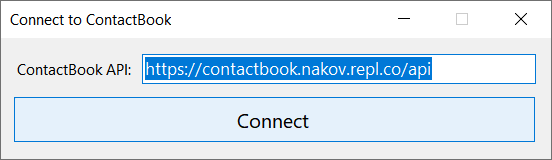


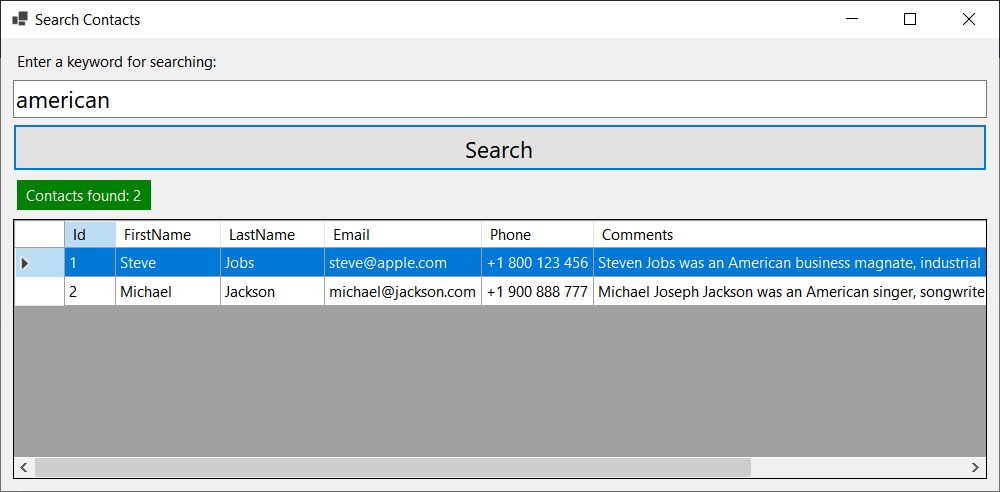
* **Navigate to "Contacts"** and assert that the first contact is "Steve Jobs" (10 score).
* **Search tasks** by keyword "albert" and assert that the first result first result holds "Albert Einstein" (6 score).
* **Search contacts** by keyword "missing{*randnum*}" and assert that there is no such contact (5 score).
* **Try to** **create a new invalid contact**, for example without name, and assert an error is returned (5 score).
* Create a **new contact**, holding valid data (first name, last name, email), and assert that the **new contact is added** and **listed last** in the contact book (14 score).

## Windows Desktop Client

Contact Book has a Windows desktop client app, available from your resources archive.

The app **connects** to the ContactBook API, **searches for contacts** and **creates new contact**:

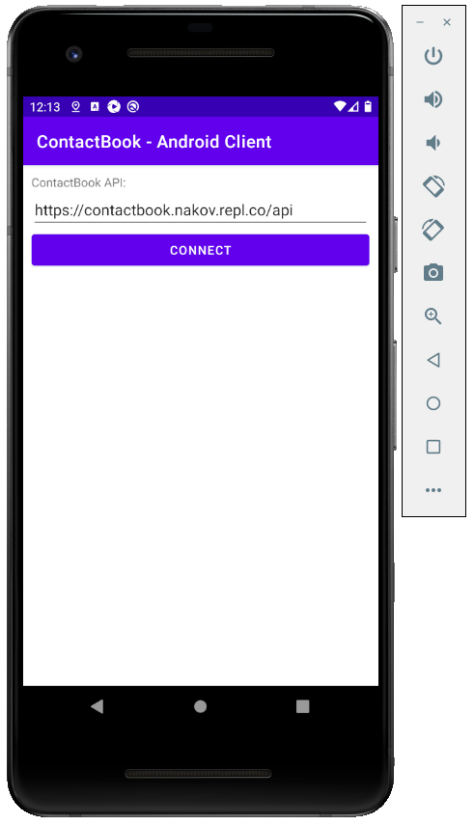
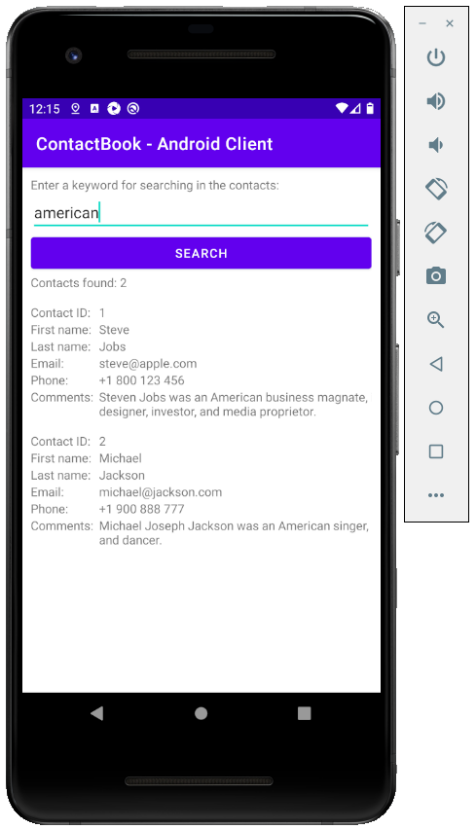
[](https://user-images.githubusercontent.com/1689586/108594524-2e527d00-7383-11eb-8958-a5a1ce0b5e29.png)

[](https://user-images.githubusercontent.com/1689586/108594544-47f3c480-7383-11eb-8b36-38cc05c77729.png)

## Android Mobile Client

Contact Book has an Android mobile client app, available from your resources archive.

The app supports **searching for contact**, using the ContactBook API:

## Appium Tests

Choose one of the next two problems: **Android app UI tests** or **Windows app UI tests**.

## 2. Contact Book Desktop App: Automated Appium UI Tests

Write **Appium-based automated Windows UI tests** for the "**Contact Book**" Windows desktop app. Implement the following automated testing **scenario** (30 score):

**Search and assert a contact**

* Open the app.
* Connect to your backend API service. The default URL address is **https://contactbook.nakov.repl.co/api, you have to change it to yours, that is part of the task,** for example: **http://{yoursite}/api**
* Switch to the new window.
* Search for "steve".
* Assert that the returned result holds first name "Steve", last name "Jobs".

## 3. Contact Book Mobile App: Automated Appium UI Tests

Write **Appium-based automated mobile UI tests** for the "**Contact Book**" Android mobile app. Implement the following automated testing scenario (30 score):

**Assert the name of the first task**

* Open the app.
* Connect to your backend API service. The default URL address is **https://contactbook.nakov.repl.co/api, you have to change it to yours, that is part of the task /**for example: **http://{yoursite}/api**
* Search for "steve".
* Assert that the returned result holds contact "Steve Jobs".

You are free to implement and run the tests in a local instance of Appium with Android Emulator or physical Android device.