Contacts Organizer SE430

Overall Description

This is a simple java desktop application for the organization of contacts. The contact details will stored and retrieved using this application. Moreover, general features, like editing, exporting, and importing contacts, are already implemented.

Software requirements specification

|  |  |
| --- | --- |
| ORGANIZATION | |
| Project name | Contact Organizer SE430 |
| Version | 1.0 |
| Organization | JUST/SE430 |
| Date created | Second semester - 2019/2020 |

Installation instructions

The source code of the Contacts application is uploaded to GitHub.

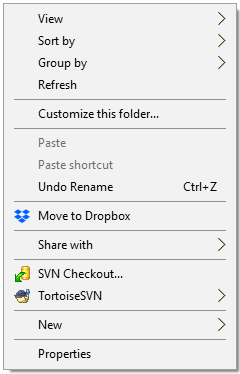
You can use Git or checkout with SVN using the following web URL:

<https://github.com/Gillou100/Java-Contact-App>

For instance, we use **TortoiseSVN** to check the source code as follows:

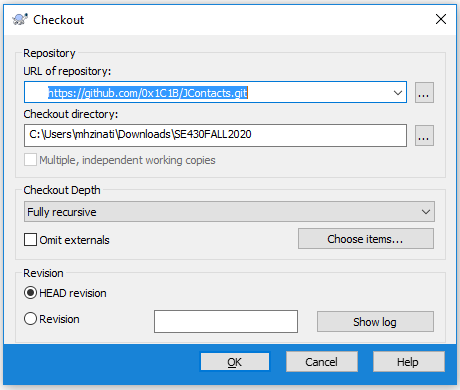
Note: if you don’t have the **TortoiseSVN** client installed in your machine, you can download it from <https://tortoisesvn.net/downloads.html>).

1. Navigate to the folder where you want to save the source code and right-click on any empty place. Click on **SVN Checkout…**

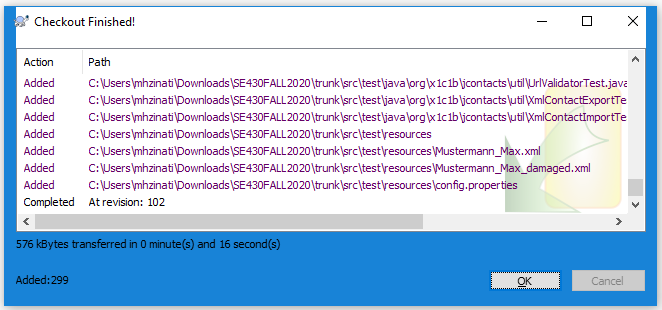


Click here

1. Enter the URL (<https://github.com/Gillou100/Java-Contact-App>) in the **URL of repository** field as shown below and click **OK**.

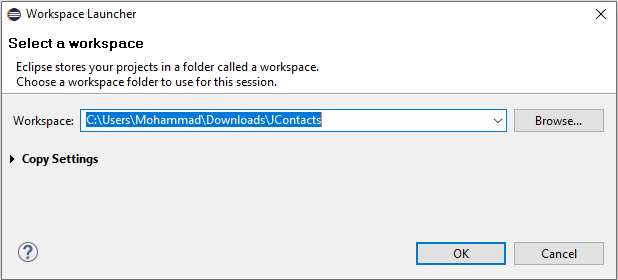


1. Wait until Tortoise finishes downloading the source code to your local machine and press **OK**.

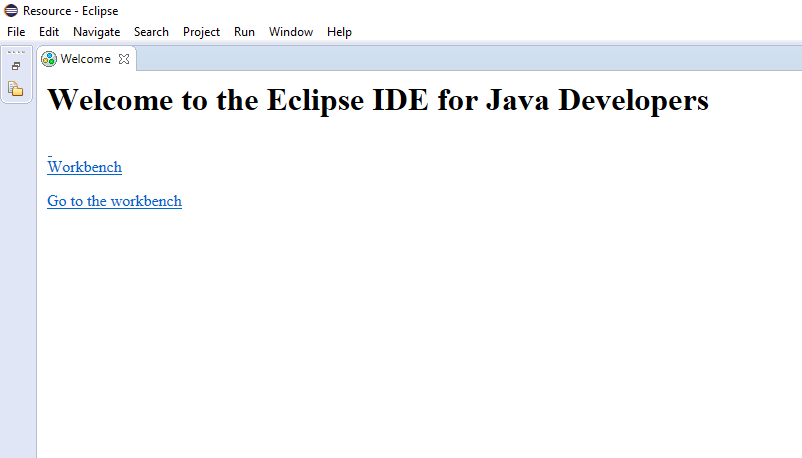


1. Start **Eclipse** and choose the workspace where you want to work. If you don’t have the Eclipse IDE for Java developer installed in your machine you can download it from the following URL

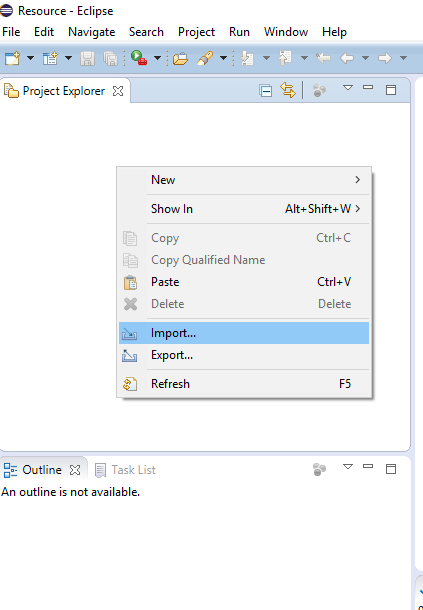
<https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers>



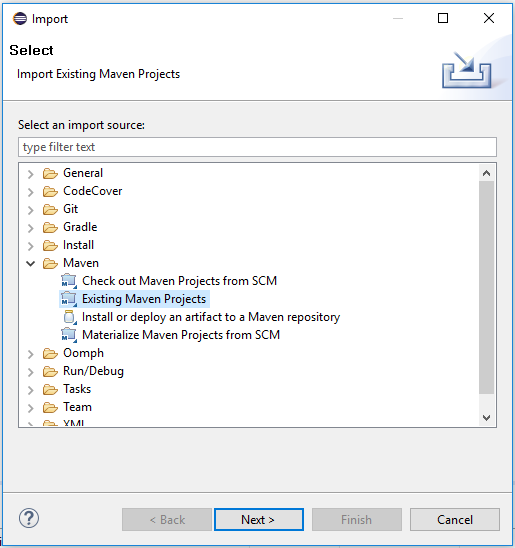
1. **Eclipse** should display the **Welcome page**. Close this page and empty **Project explorer** will be displayed.



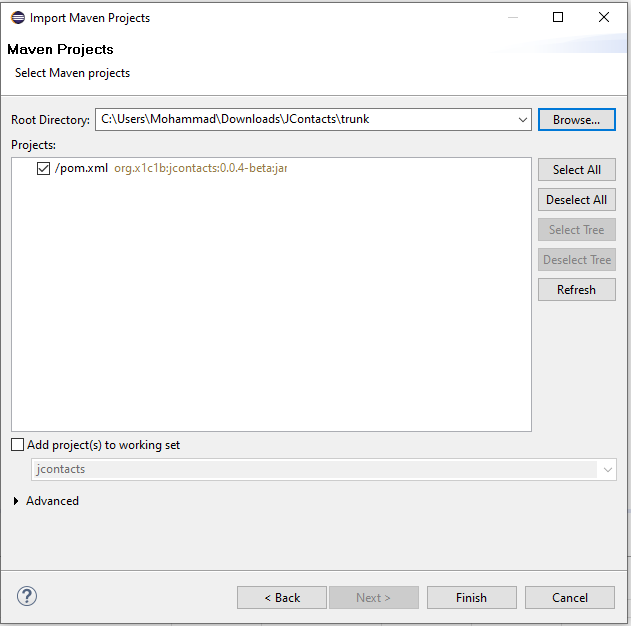
1. Right-click on the empty project explorer and click on **Import…** option.



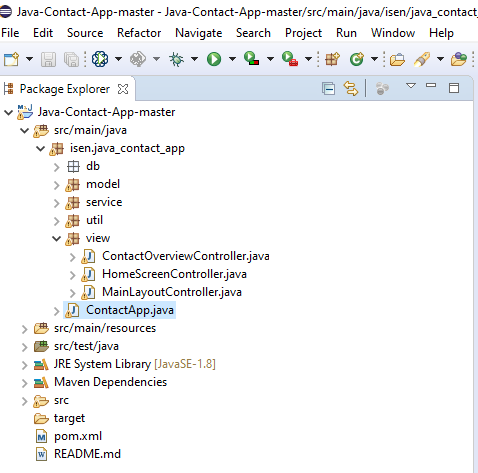
1. Choose to import (**Existing Maven Projects**). **Apache Maven** is a software project management and comprehension tool. In case you are interested in more information about this tool, please visit the **Maven** website at <https://maven.apache.org/>.



1. Choose the parent folder. The **pom** files will be automatically detected. Just press **finish** and the project will be imported.

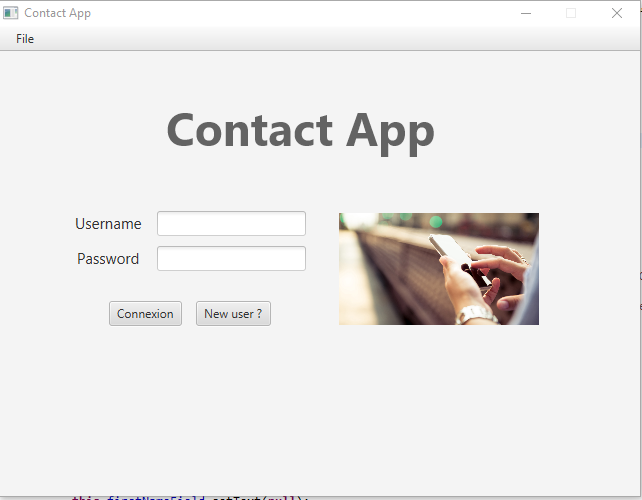


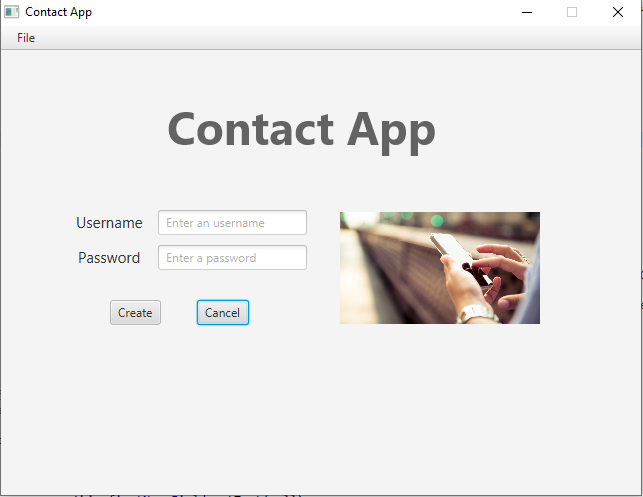
1. Wait until eclipse finishes importing your project. Your project explorer should look like the following:



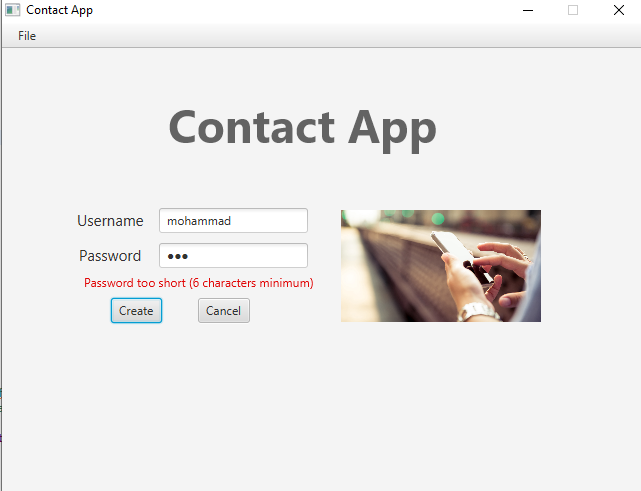
1. Your project is now ready to be executed. Right-click on **ContactApp.java** and choose **Run As => Java application** and the ContactApp application will start running.

To use the Contact App, you should create a user with a valid password with a minimum of 6 characters length. This is done by clicking on the “New User ?” button.

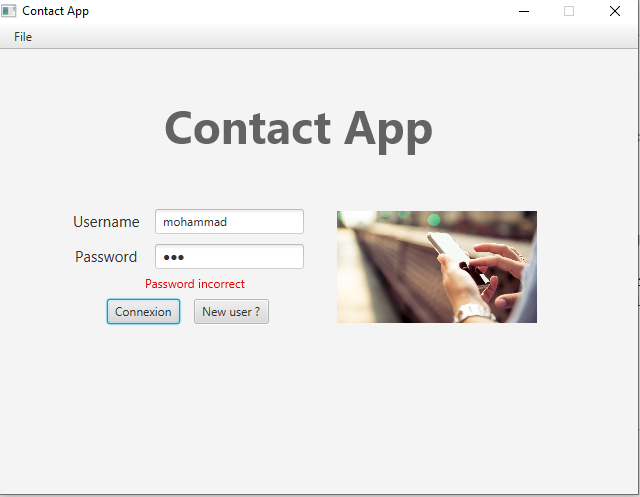




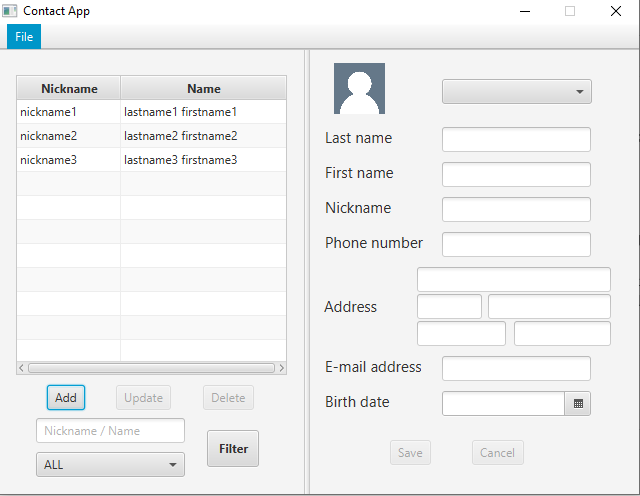
The user should enter a user name with a password and press “Create”



In case the password is short, the application should show an error message. Otherwise, if the user is created correctly, the application should display its main window.



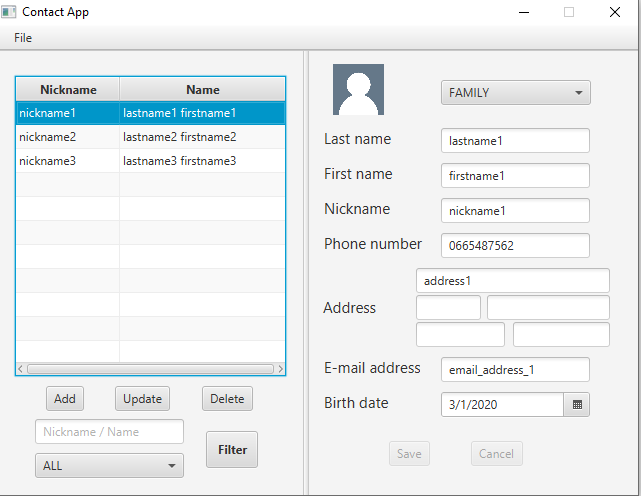
In case the user has a username, he can skip creating a new user account.



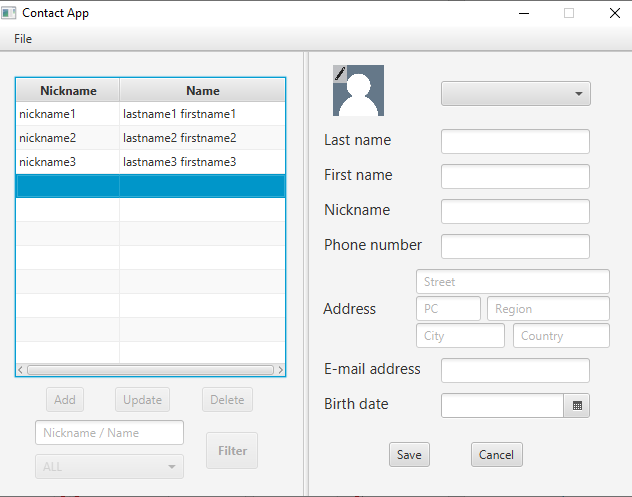
**Contact List area**

**Contact details area**

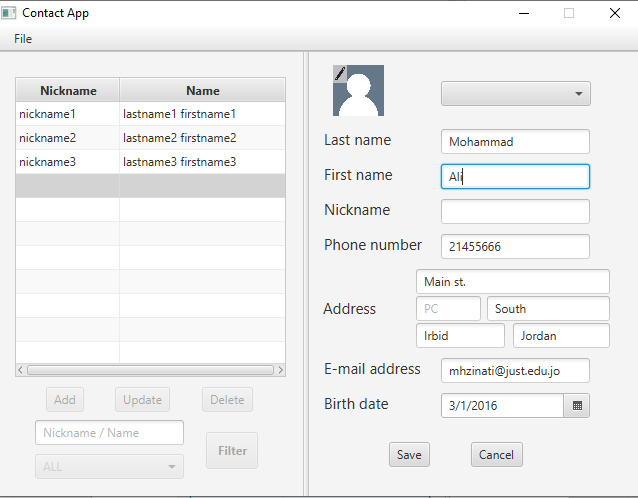
The contact app main window is displayed after creating a new username or after entering the correct username with its password. Initially, the contact list area is enabled and the contact details area is disabled. Besides, the update and delete buttons are initially disabled.



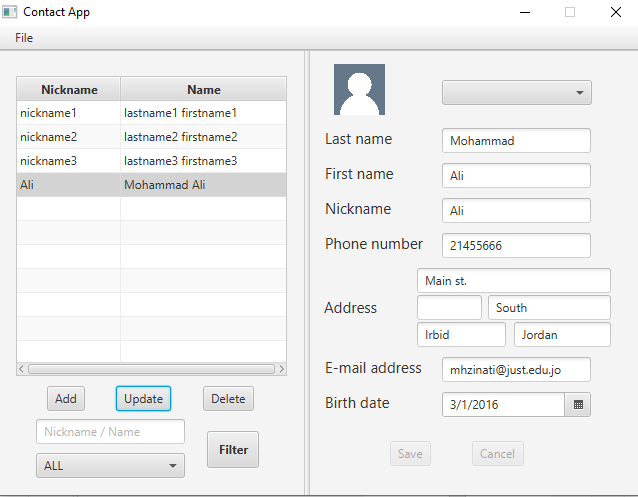
When the user selects a contact on the list, the update and delete button is enabled and the contact details area is populated with the selected contact details.



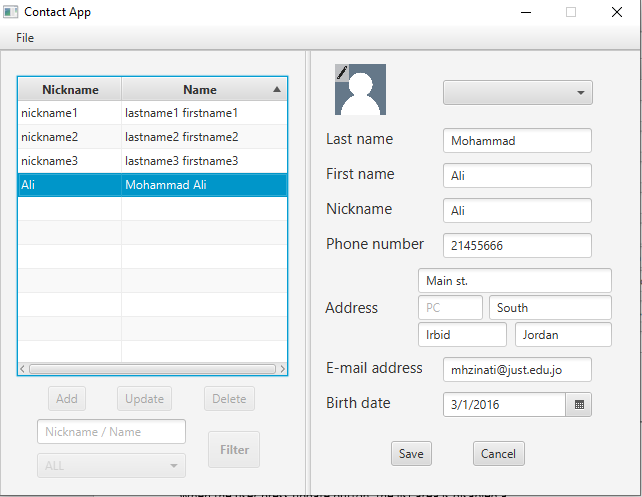
When the user presses add button, a new record is created on the contact list area and the contact list area is disabled. At the same time, the contact details area is enabled.



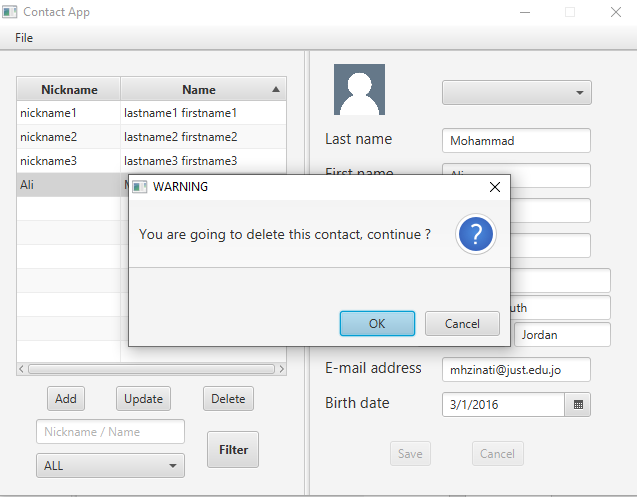
When the user press the save button, the contact is added to the list. The contact details area returns to be disabled and the contact list area is enabled.



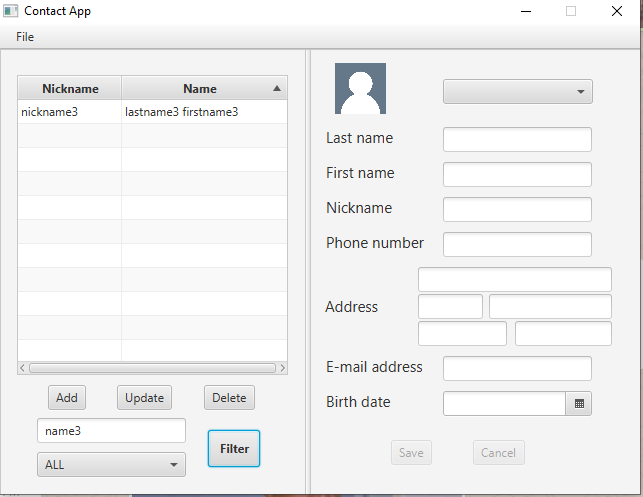
When the user press update button, the list area is disabled and the details area is enabled and populated with the selected contact details.

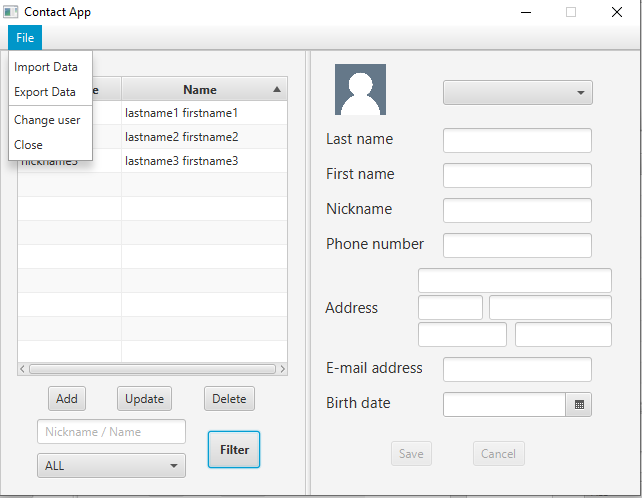


When the user press the delete button, a warning message is displayed to the user to confirm if he really wants to delete the contact. If the user press cancel, the contact will not be deleted. Otherwise, if the user press the OK button, the user will be removed from the list.

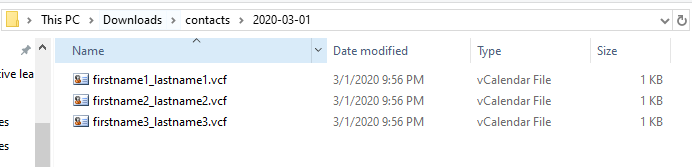


The user can enter part of the name / nickname in the filter field and press the filter button. The app should display the list of contacts that satisfy the search criteria.

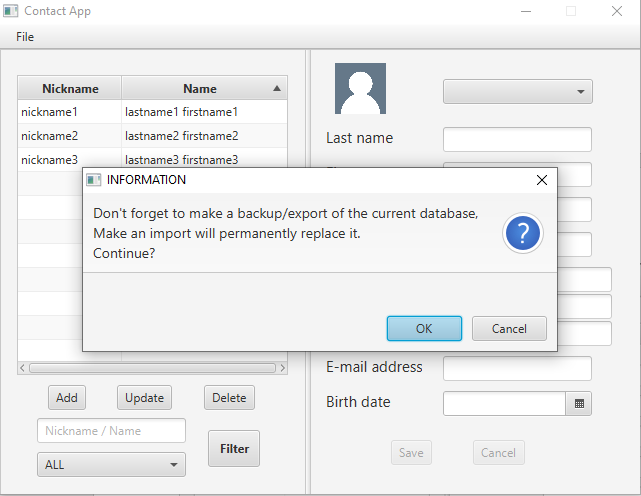




The user can select Export Data from the File menu. The app should export each contact in the list in a separate file.



The user can select Import Data from the File menu. The app should display a message to confirm that import will replace existing data. The imported contact should be displayed in the list after that.



Finally, the user can select Change user from the file menu. In this case, the main login menu should be displayed.

Deliverable (1) Description

Using the black box testing techniques that we learned in this semester, you are required to design test cases and execute them to test the Contact application. You should submit a report that contains the following:

* 1. The black box testing technique (s) that you used to achieve the given task. You should also write a paragraph to justify your choice.
  2. The set of test cases and the results of their execution.
  3. The set of bugs found during the testing process.