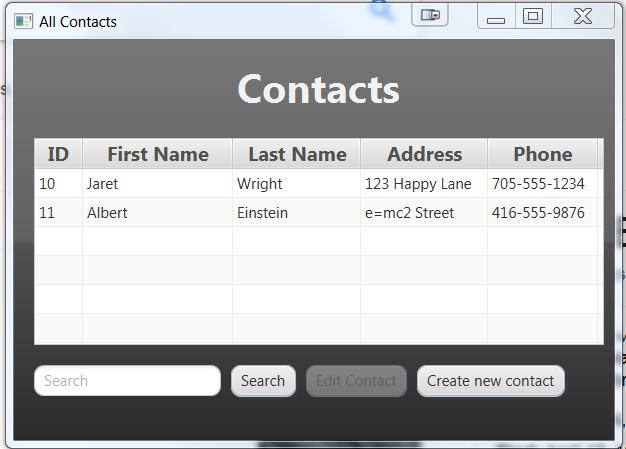
# Assignment 1

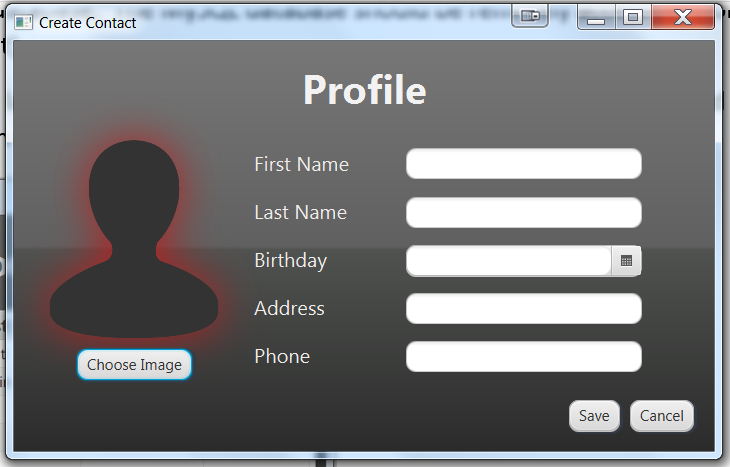
## Overview

The goal of assignment 1 is to create a graphical user interface that will allow users to define and save “contacts” to a MySQL database. The MySQL database should be remotely accessible (for example use your Dream host account).

When the application is launched, it should show a table of contacts. Your profile should be in the first position of the table. An example of how the contacts page could look is shown below.



If the user clicks on the “create new contact”, the application should change scenes and provide the user with the ability to add information about the contact. The create scene should have a default image and have fields to fully define a contact. The fields shown below are the minimum fields that should be in the scene. Other options could include radio buttons for male/female, a notes section, email address, etc.…



When the user populates the form and hits save, the information should be stored in a database.

## Requirements

1. When the application launches, it takes the user to a scene with a table prepopulated with at least 3 contacts. The first contact should be you (details do not need to be real).
2. When the user clicks on a button to “create new contact”, they should be taken to a scene that allows them to enter information for the contact.
3. When the user enters invalid information in the create scene, they should be notified by displaying what went wrong. For some fun, try changing the scene so that the area of concern shows with a red outline (Hint: use CSS for this).
4. If all the information in the form is valid, pushing the “save” button should create a new instance of a contact and save it in the database. The system should then change scenes to the table of all contacts. The new contact should be visible in the table.

## Grading

All of your marks will be based on the rubric defined below (and visible in Blackboard).

| Criteria | Level 0 | Level 1 | Level 2 | Level 3 |
| --- | --- | --- | --- | --- |
| Style | The code does not follow typical Java programming style. I.e. a capital letter to start all class names, lower case letters start variable and method names | The code is indented and has proper upper case/lower case conventions | All of level 1, plus each method has a Javadoc style of comment prior to the method describing what it does |  |
| Requirement 1 - when launched, a TableView object is populated with contacts | The first scene is not a TableView object populated with the contacts | The first scene is a TableView object, but the contacts did not load. | The first scene is a TableView object, populated with at least 3 contacts, but they are not being read from a database. The first contact is you. | All of level 2, plus the contacts are being read from a database. |
| Requirement 2-change to create contact scene | There is no utility to change from the TableView scene to the Create contact scene | There is a button present to change scenes and pushing it takes the user to the Create Contact scene |  |  |
| Requirement 3 - error handling | There is no error handling. If an invalid input is made, the program may generate an exception and/or not inform the user | If there is an invalid input, it will display a message to the screen | All of level 1, plus the message will be useful to any user. I.e. ensure your messages are clear |  |
| Requirement 4 - create a valid contact | There is no way to create a valid contact (i.e. the save button is missing or not functioning). | Pushing the save button will create a valid contact object. | Level 1, plus the new contact will be saved in the database | Level 2, plus the user will be redirected to the TableView scene, which will be loaded with the new contact |
| Database | There is no remote database connectivity in the project and/or the necessary SQL was not provided for a local DB | There is database access, but it is local and the necessary SQL statements are missing | There is local DB access and the SQL statements are provided OR a remote DB is used and all necessary info is provided |  |
| Submission | A link to your private GitHub account was not submitted on Blackboard. | The submission was sent as a zip file that contains ALL necessary inputs to import into a Netbeans environment. | A private Github link was sent. JaretWright is listed as a collaborator AND all project files including the build info are present |  |

## Bonus

For 2 additional marks implement a search functionality on the contacts TableView. This should filter out all contacts that do not have the search term in the first or last name.