# Login page

The front-end of the application is guarded by a login page. This login page accepts 2 user input arguments: username (e-mail address) and password. The login button processes the data. The login controller will validate the user input, reroute the user to the home page within the application on success, or reroute the user back to the login page with an error.

Sketch:



Error message:



## Basic style conventions

The login page has a container with a login form. The container is centered horizontally and vertically. When resizing the browser, the box should move to the center again.

We use the boostrap standard form elements and guidelines for primary / secondary actions. Login button is primary action.

We will want to divide a form grid into 1/4 of the width for the label, 3/4 of the width for the form element.

### Language

All labels and placeholder form text should be defined as a translatable label. Like “APP\_BUTTON\_LOGIN”. The translator of the application will translate the label to the appropriate translation. The default language for the login page is ENGLISH, but must be able to change to something different quickly

In the application the contact “language\_code” is used for the user to find the language

### Auditlog

Every login attempt (successful or wrong) is written in the auditlog table. This log table has a generic purpose, so the services you write for the storage of the records will also be used elsewhere (in a use case where someone changes settings etc, removes a user, adds a company etc.).

Based on te audit log, we will be able to block an IP-address if repeated wrong login input is given (detecting intrusion).

### Userdata

A user is found in the database by the relation of the user table and the contact table (user.id inner joined on contact.user\_id).

For now, a manual entry in the database of a single contact and a corresponding user will be fine for the this function. Later on, we may need a seeder to populate data in the database.