Functional specification

Bug-app v2.0 (Sandbox castle)

Table of contents

[Table of contents 2](#_Toc379902120)

[Table of figures 4](#_Toc379902121)

[1 Overview 5](#_Toc379902122)

[1.1 Purpose 5](#_Toc379902123)

[1.2 Vision 5](#_Toc379902124)

[2 Requirements 8](#_Toc379902125)

[2.1 User model change 8](#_Toc379902126)

[2.2 Authentication system 8](#_Toc379902127)

[2.3 New user profile page 8](#_Toc379902128)

[2.4 Editing user profile 8](#_Toc379902129)

[2.5 Smart email and account validation 8](#_Toc379902130)

[2.6 Personal data hiding 8](#_Toc379902131)

[2.7 Link for user profile 9](#_Toc379902132)

[2.8 Messaging system 9](#_Toc379902133)

[3 User interface 10](#_Toc379902134)

[3.1 Profile page 10](#_Toc379902135)

[3.2 Registration page 10](#_Toc379902136)

[3.3 List page 11](#_Toc379902137)

[3.4 Index page 12](#_Toc379902138)

[3.5 Login page 13](#_Toc379902139)

[3.6 404 page 13](#_Toc379902140)

[4 Validation 15](#_Toc379902141)

[4.1 Email and account validation 15](#_Toc379902142)

Table of figures

[Figure 1 profile page prototype 1 6](#_Toc380150302)

[Figure 2 profile page prototype 2 6](#_Toc380150303)

[Figure 3 registration page 10](#_Toc380150304)

[Figure 4 registration page edit controls 10](#_Toc380150305)

[Figure 5 registration page ok and no buttons 11](#_Toc380150306)

[Figure 6 registration page 12](#_Toc380150307)

[Figure 7 list page 13](#_Toc380150308)

[Figure 8 list page elements 14](#_Toc380150309)

[Figure 9 index page 14](#_Toc380150310)

[Figure 10 login page 15](#_Toc380150311)

[Figure 11 404 page 15](#_Toc380150312)

[Figure 12 email and account smart validation 17](#_Toc380150313)

# Overview

## Purpose

Current application functionality is satisfying the business needs. Although there are some critical issues with the production environment. The main points are:

1. sometimes users input incorrect data and realize it only after submission when he cannot change them
2. currently to change user data manual database handling is required; running bare SQL scripts on the production database is hard-to-perform operation which cost
3. some personal data can be seen by strangers which is not acceptable

In order to handle these problems this document was created.

## Vision

First thing to do is to add user data editing functionality. This can be a dedicated page which will mirror the data on the list page and permit to edit it.

Dedicated page (profile page) can also contain user personal data like his birth date and his own description about self. Also a placeholder for user photo would be nice to see. All in all the page should look like on the picture.

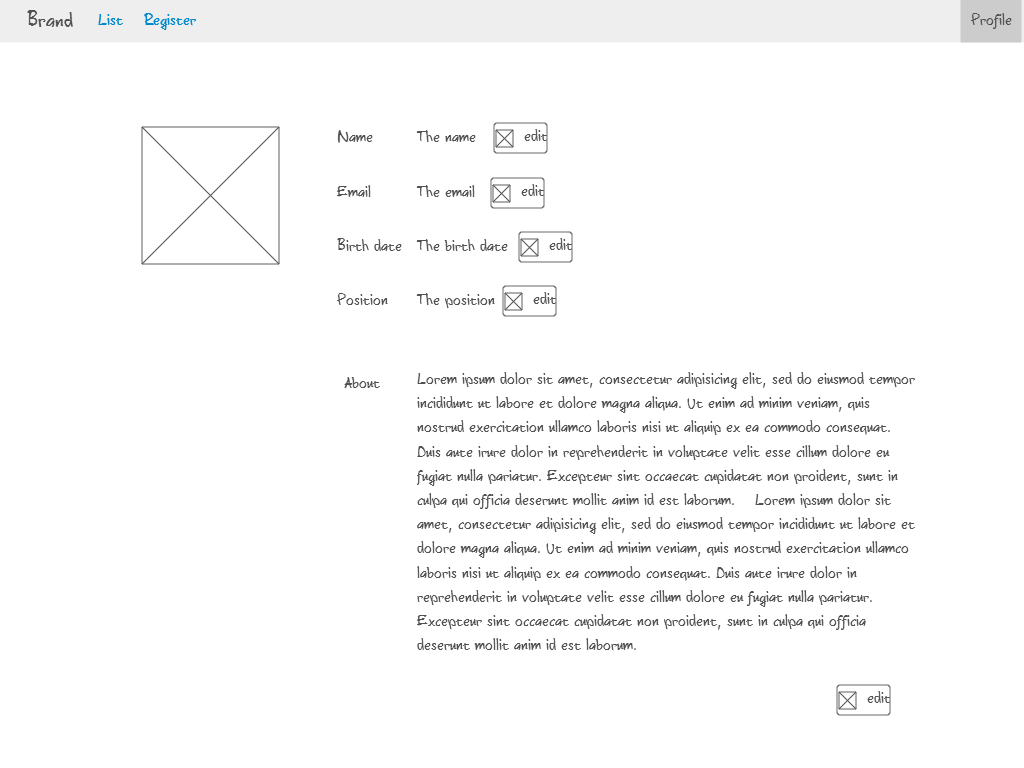


Figure profile page prototype 1

To edit a field a special button near will be usable, after clicking on it editing functionality should become available. For data field a datapicker will be nice, also dropdown element for position field.

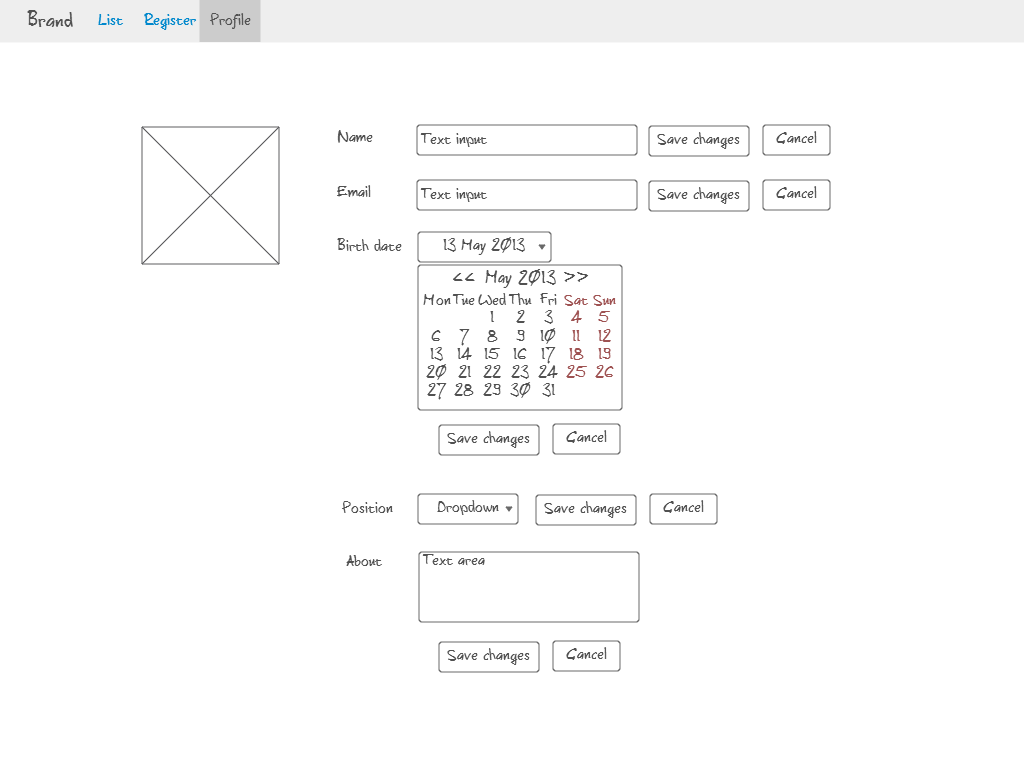


Figure profile page prototype 2

Also basic authentication mechanizm can solve personal data problem. We can hide it under profile page and demand user authentication to open. User should be able to edit only it's own profile of course, other profiles are read-only.

Email is not an acceptable key for user login, another one (like account or username) should be presented. We need to use these accounts for our LDAP system and email is not a perfect candidate for some LDAP connected systems.

# Requirements

## User model change

User information must be expanded with new fields which are:

* account name
* last name
* password

New fields must be presented on registration page and be validated. There must be no password strength check. Check for already existing account name and email should be presented.

## Authentication system

Basic authentication system must be implemented. New user field account must be used as a login. There are no requirements to a password field, it can be any. Only authenticated users have permission to open profile page.

## New user profile page

New page must be created where user information can be edited. The new fields account and last name and must be there as well. User cannot edit his sex or password. User can edit only his own profile, other profiles for him are read-only.

## Editing user profile

For editing user data on profile page validations performed must be all the same as on registration page. There must be controls to edit user data near each field. About field is not filled on registration page so it is allowed to be empty.

## Smart email and account validation

Email and account fields must be validated by server – check if such email or account is already registered.

## Personal data hiding

Birth date and about fields should not be presented on user list page. They are considered as personal data and should be shown only on profile page after authentication.

## Link for user profile

On the users list page under account column there must be a link to a user profile.

## Messaging system

Messaging system must be implemented. Messages must be shown in the right top corner on each page. Messages are shown after a successful registration or successful user data change and other places. Messages should appear as pop-ups with text and a close button.

## Cyrillic and umlauts support

Cyrillic and umlauts support must be introduced.

## Security

Application must satisfy basic security concepts described in [ch.5](#_Security).

# User interface

## Profile page

Profile page can be accessed only if user is authenticated. Here is presented all user information including birth date and his about section. A user photo placeholder must be on the left part of the page, for now just a template.

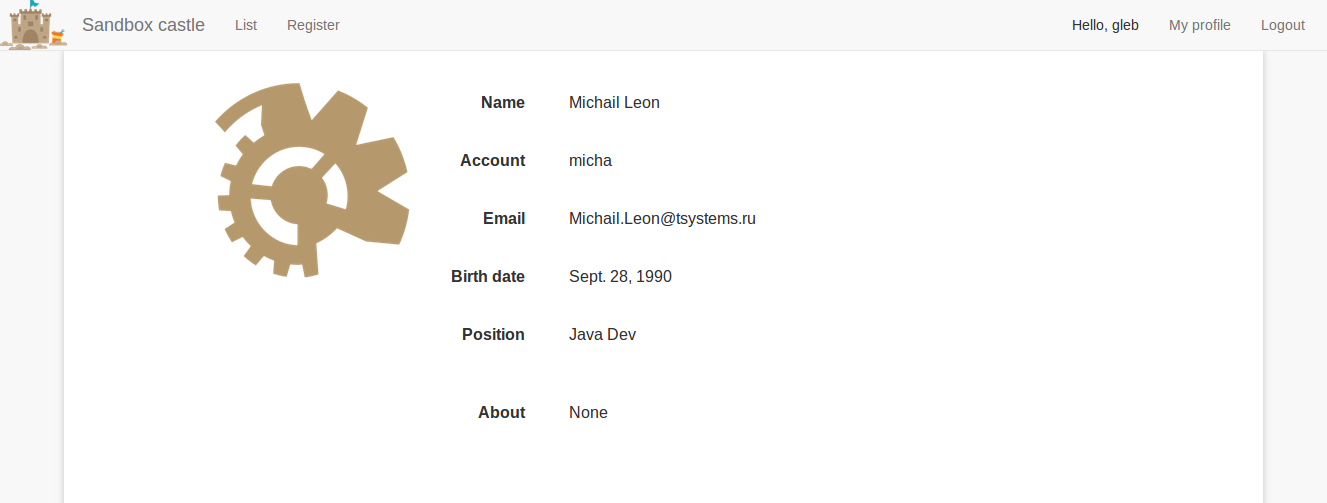


Figure 3 registration page

If user is looking onto his own profile, edit controls appear near each field allowing him to edit this field.

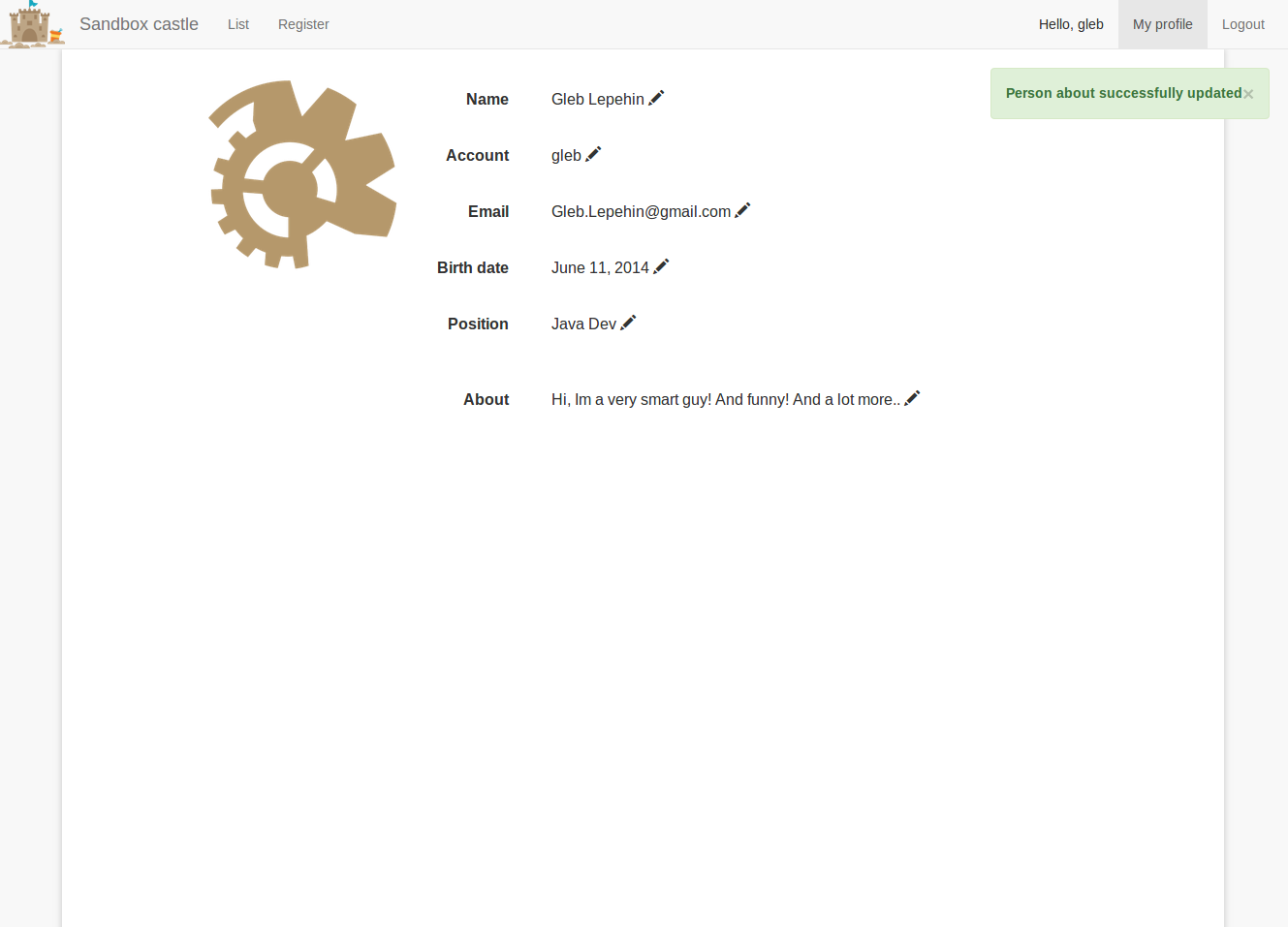


Figure 4 registration page edit controls

After clicking on the edit control the element must be transformed into input element. Also 2 buttons must be presented: ok – to accept changes and send them to server, no – do decline changes and return the field to it’s starting value.

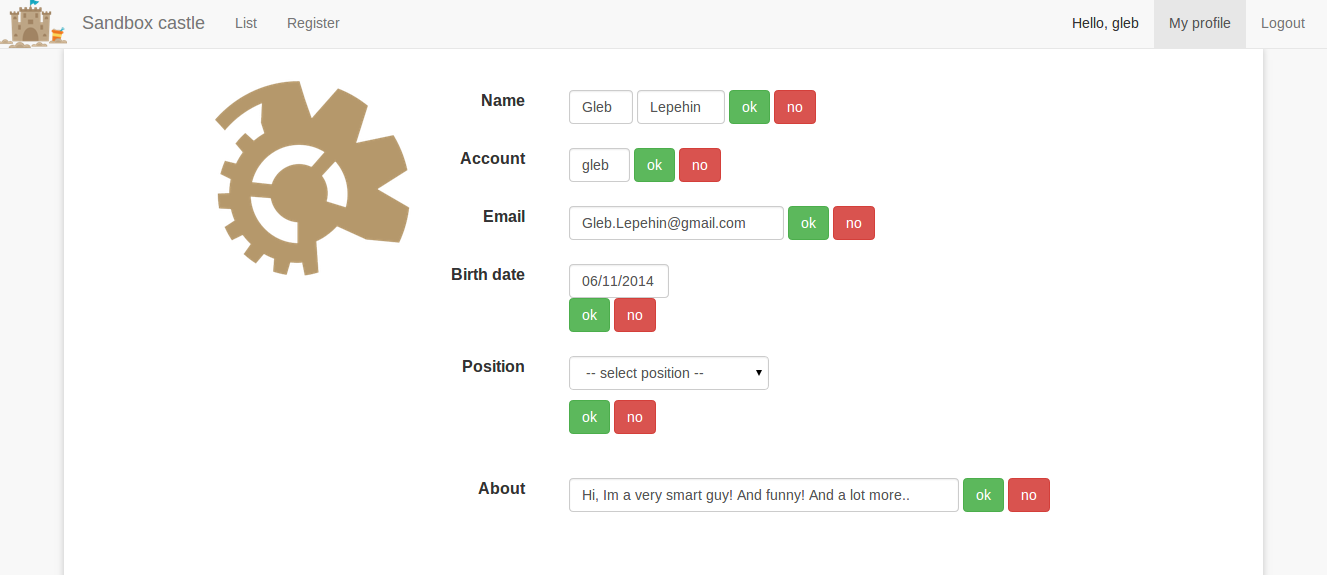


Figure 5 registration page ok and no buttons

After changing some field successful message in the right top should appear.

All validations on the profile page should be the same as on registration page (see [3.2](#_Registration_page)).

## Registration page

The registration page should look as follows.

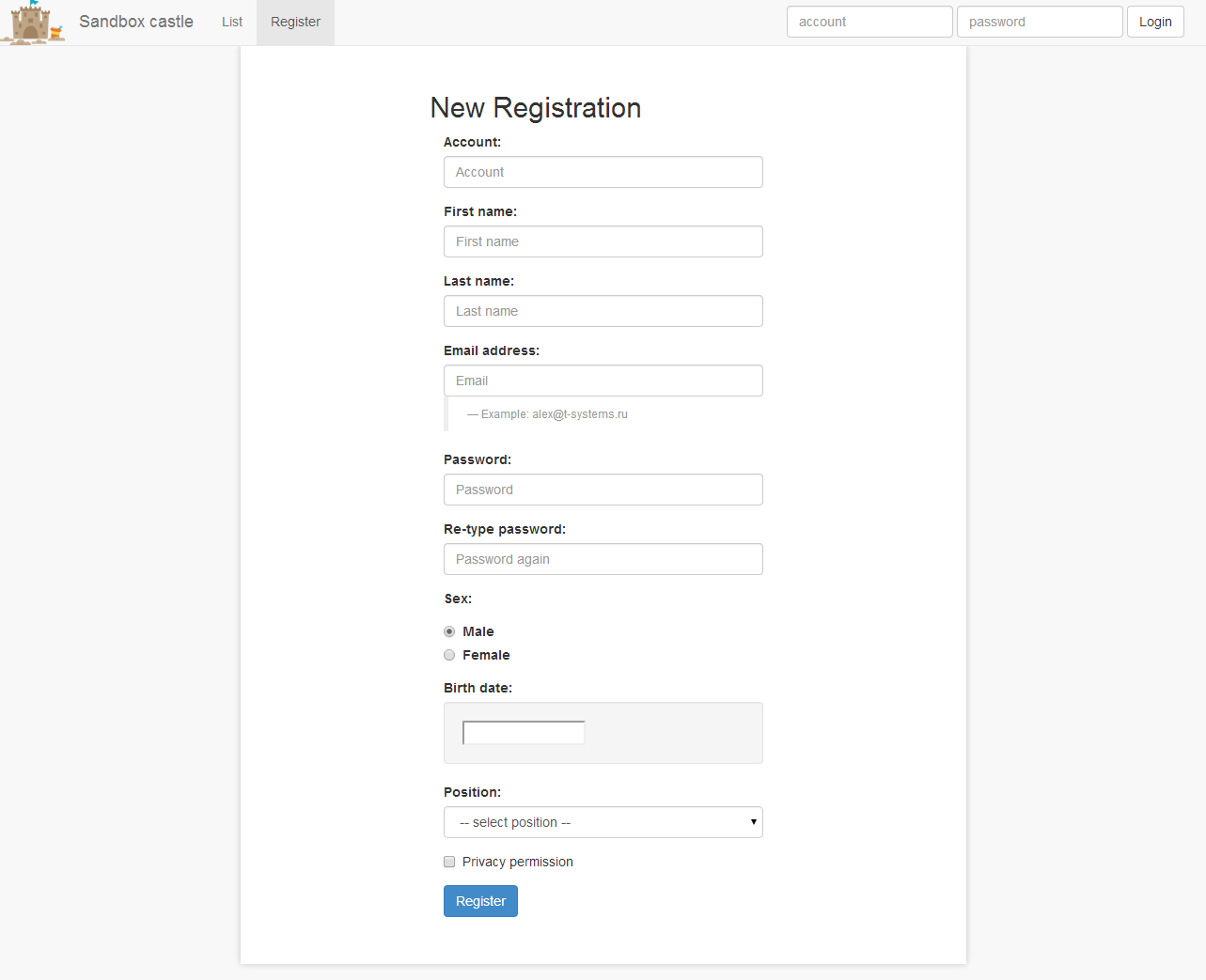


Figure registration page

Form must be at the center of the page, input field label should be above the element. On the bottom of form there must be a submit button.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Type | Value | Validation |
| **Account** | input field (text) | user input | mandatory, maxlength=30, smart check (see 4.2) |
| **First name** | input field (text) | user input | mandatory, maxlength=20 |
| **Last name** | input field (text) | user input | maxlength=20 |
| **Email** | input field (text) | user input | mandatory, maxlength=40, smart check (see 4.2) |
| **Password** | input field (text) | user input | mandatory, maxlength=20 |
| **Re-type password** | input field (text) | user input | mandatory, maxlength=20, password equality check |
| **Sex** | input field (radio) | user select | mandatory |
| **Birth date** | input field (text with datepicker) | user select | mandatory, format check “mm.dd.YYYY”, maximum range is current date minus 100 years |
| **Position** | input field (select) | user select | mandatory |
| **Privacy permission** | input field (checkbox) | user select | mandatory |

Registration must not proceed if any of fields is not valid.

## List page

On the list page there must be a list of all registered users. Only the most important user data is shown here. All hidden data can be seen on profile page.

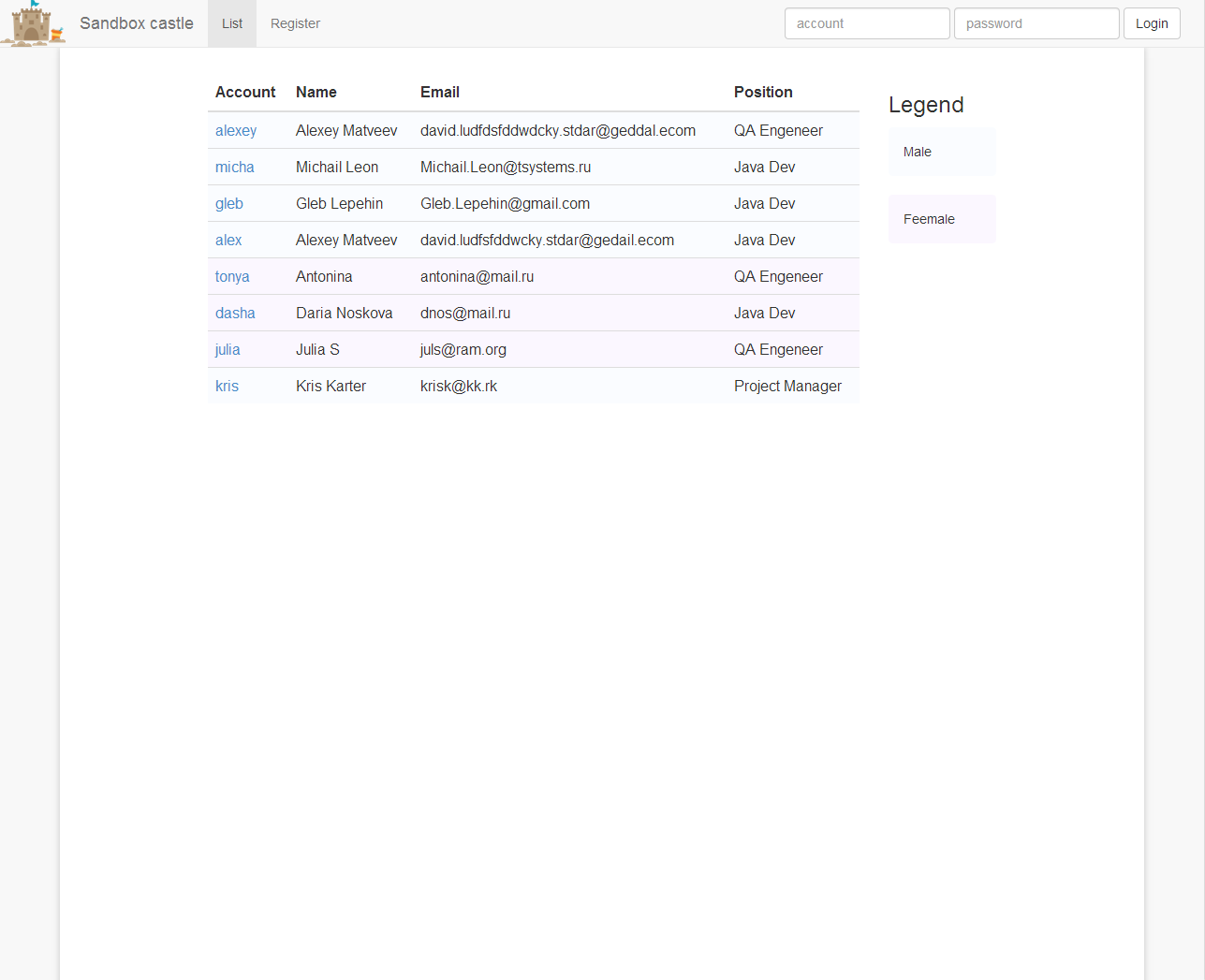


Figure list page

Legend is an additional widget that helps users to understand the meaning of list items. Currently the only legend item is sex separation using colors. Table row representing user should be colored according to the user sex.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Type | Value | Validation |
| **List of employees** | table | - | - |
| **User account** | column | “Account” | - |
| **User name** | column | “Name” | - |
| **User email** | column | “Email” | - |
| **User position** | column | “Position” | - |
| **Legend label** | label | “Legend” | - |
| **Male/female legend** | block | “Male”, “Female” with colors | - |

Figure list page elements

## Index page

Index page must open clicking on the brand logo. It currently have a template data reserved for further releases.

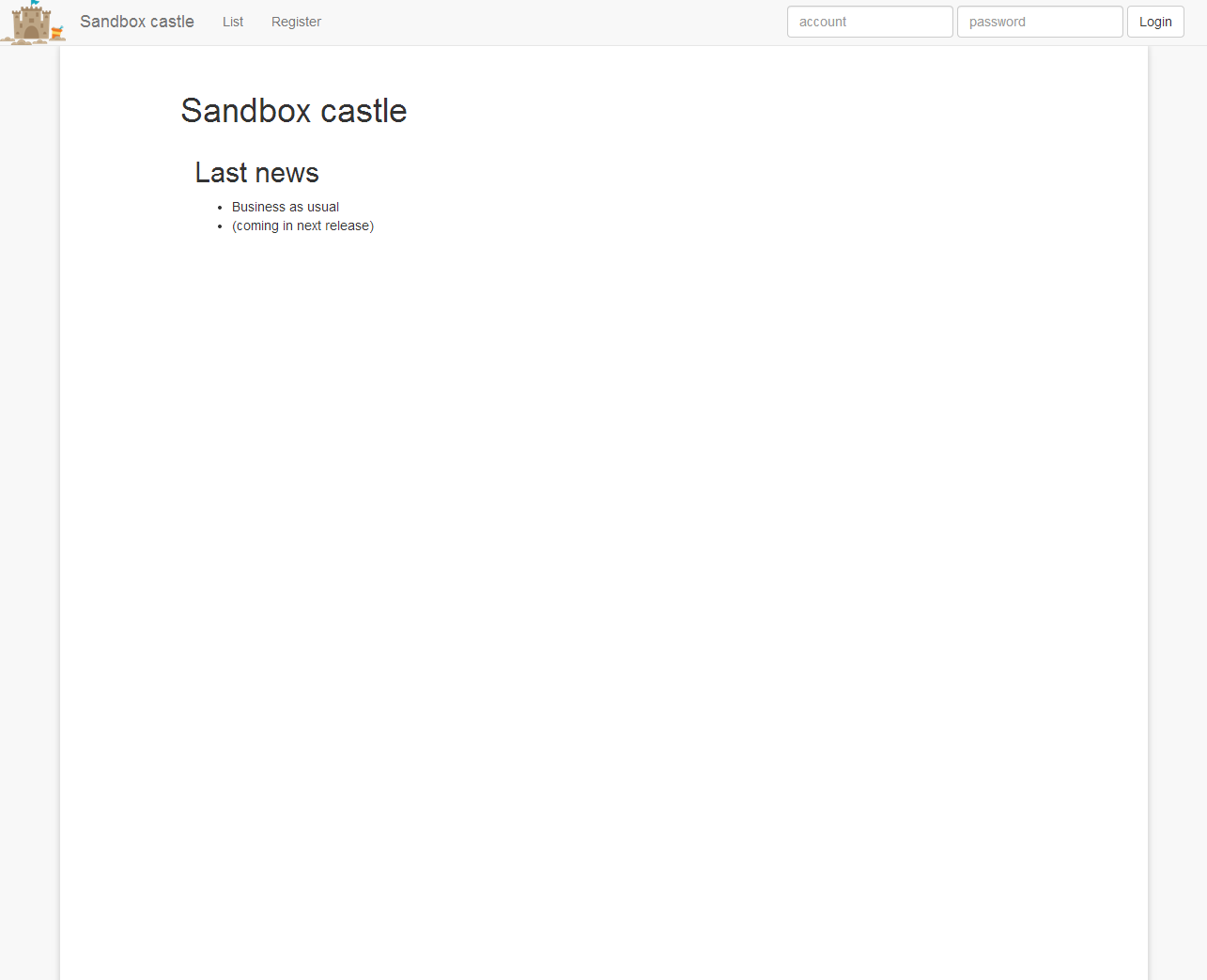


Figure index page

## Login page

Login page must appear if non-authenticated user is trying to access protected system places. This page is an addition to login controls in the main menu’s right corner (that are available all the time). Current list of protected pages that require authentication:

* profile page

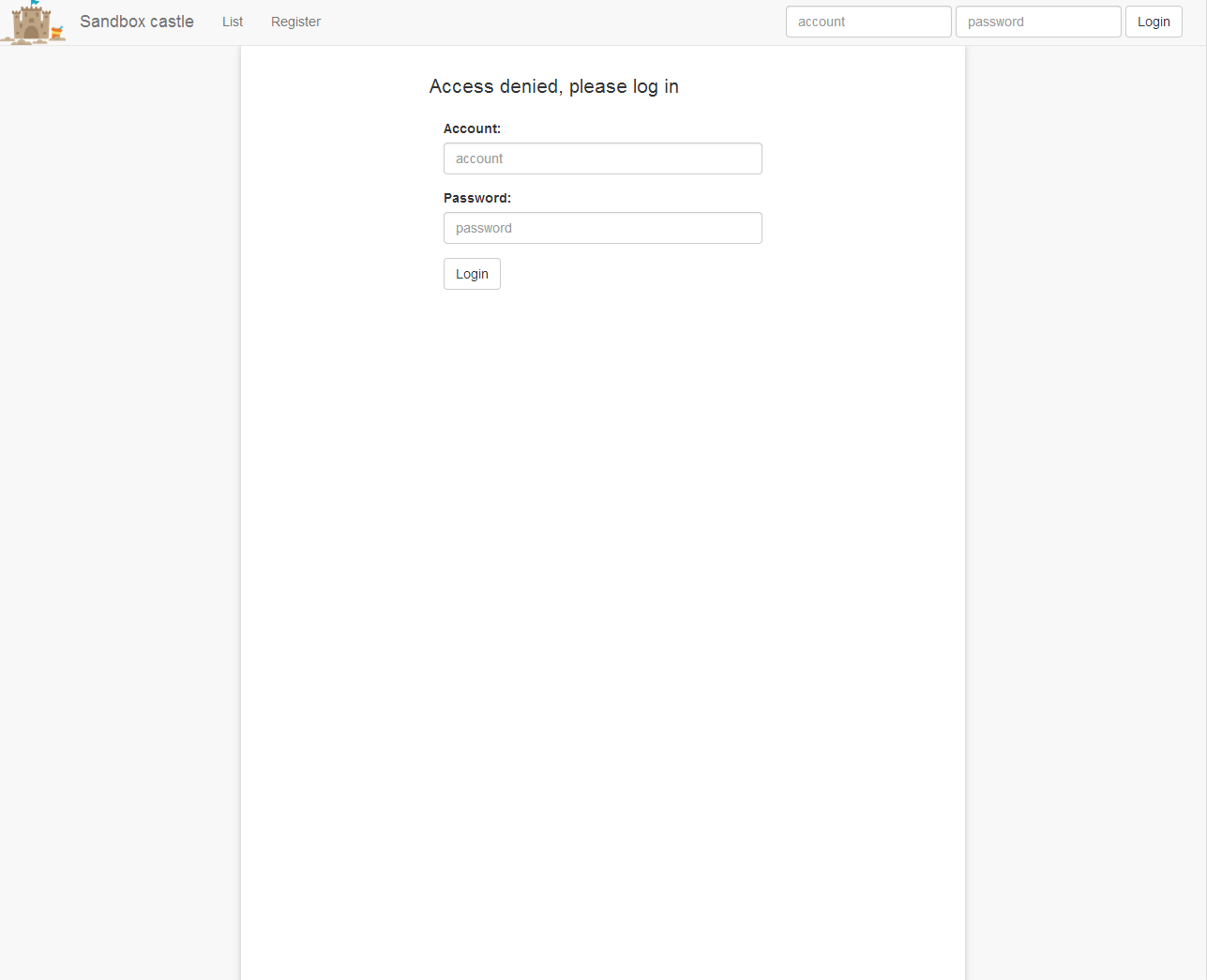
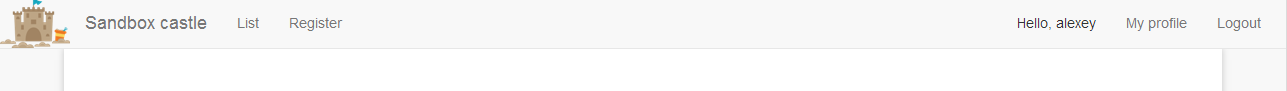


Figure login page

After successful authentication login controls are replaced with a menu button that links to a user profile page and a logout button.



## 404 page

Default browser 404 page must look as follows.

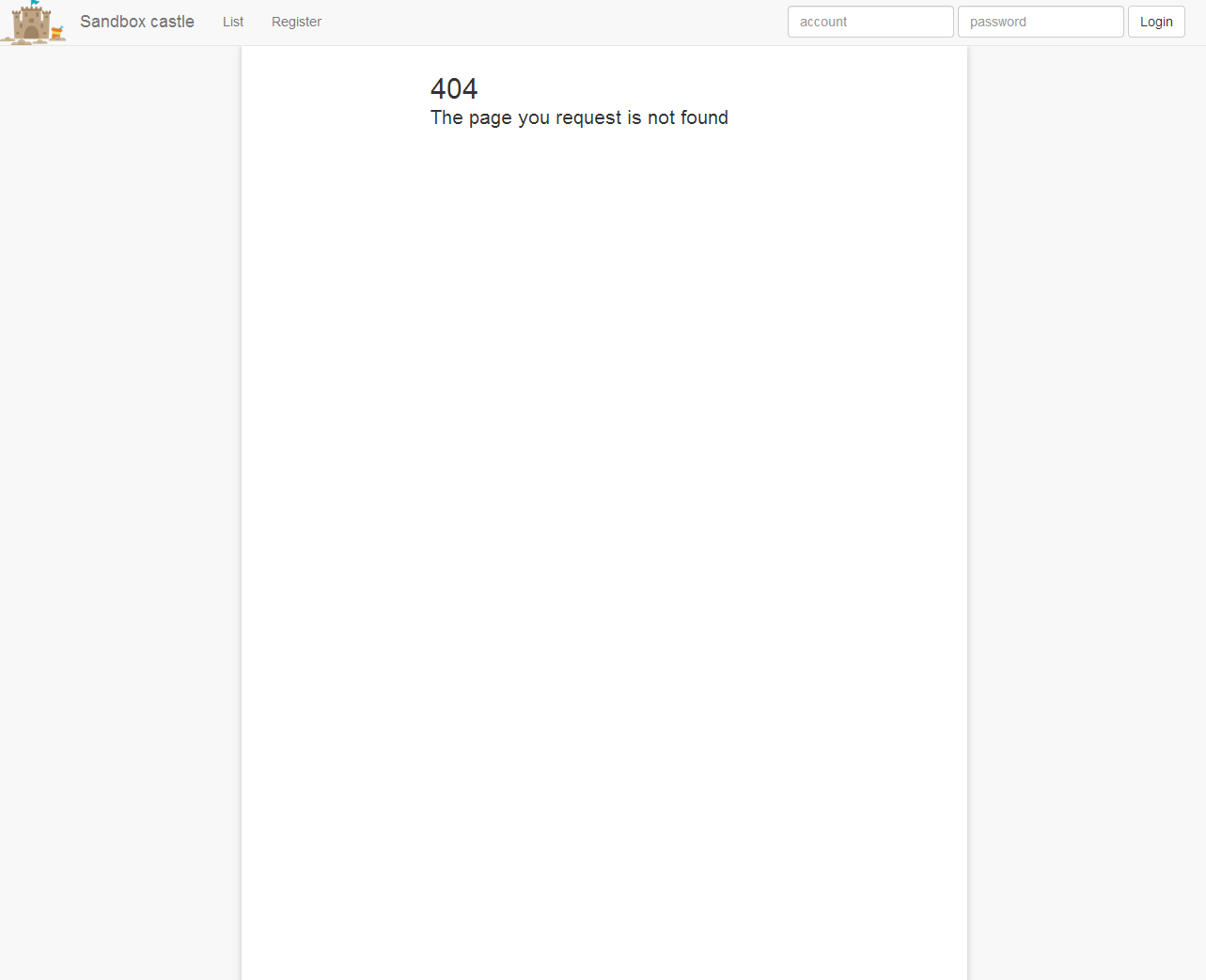


Figure 404 page

Any call to a page that server does not know must be redirected to 404 page. All server errors must be revealed with a message in the right top.

# Validation

## Email and account validation

Smart validation for account and email fields must be presented. During input asynchronous call must be performed to the server to check if currently typed account (email) is already registered. If not – a special validation message must be shown.

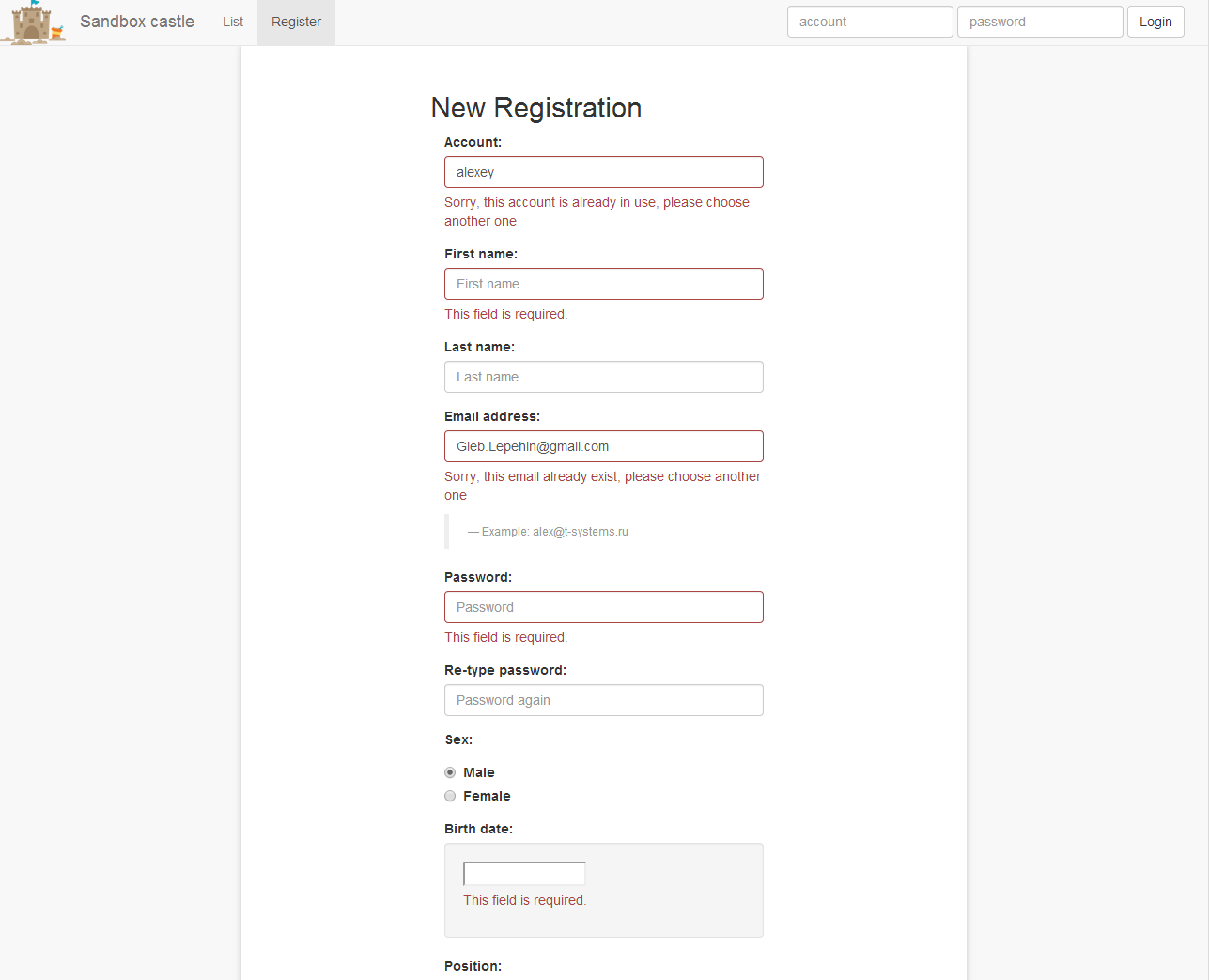


Figure email and account smart validation

# Security

## Security

Server side must not be vulnerable to SQL injection or other client-side attack (like XSS). All unknown pages must throw 404 error and redirect to 404 page (see [3.6](#_404_page)).

The available pages are:

* %/ - [index page](#_Index_page)
* %/list - [list page](#_List_page)
* %/add - [registration page](#_Registration_page)
* %/user/<username> - [profile page](#_Profile_page)
* %/admin - admin portal (for admin only)

, where % = http://<host>:<port>

There are no requirements on server load as maximum of 500 users are expected per day.