Functional specification

Bug-app v2.0 (Sandbox castle)

Table of contents

[Table of contents 2](#_Toc379758601)

[Table of images 3](#_Toc379758602)

[1. Overview 4](#_Toc379758603)

[1. Purpose 4](#_Toc379758604)

[2. Vision 4](#_Toc379758605)

[2. Requirements 6](#_Toc379758606)

[3. User interface 7](#_Toc379758607)

[4. Validation 8](#_Toc379758608)

Table of images

# Overview

## Purpose

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

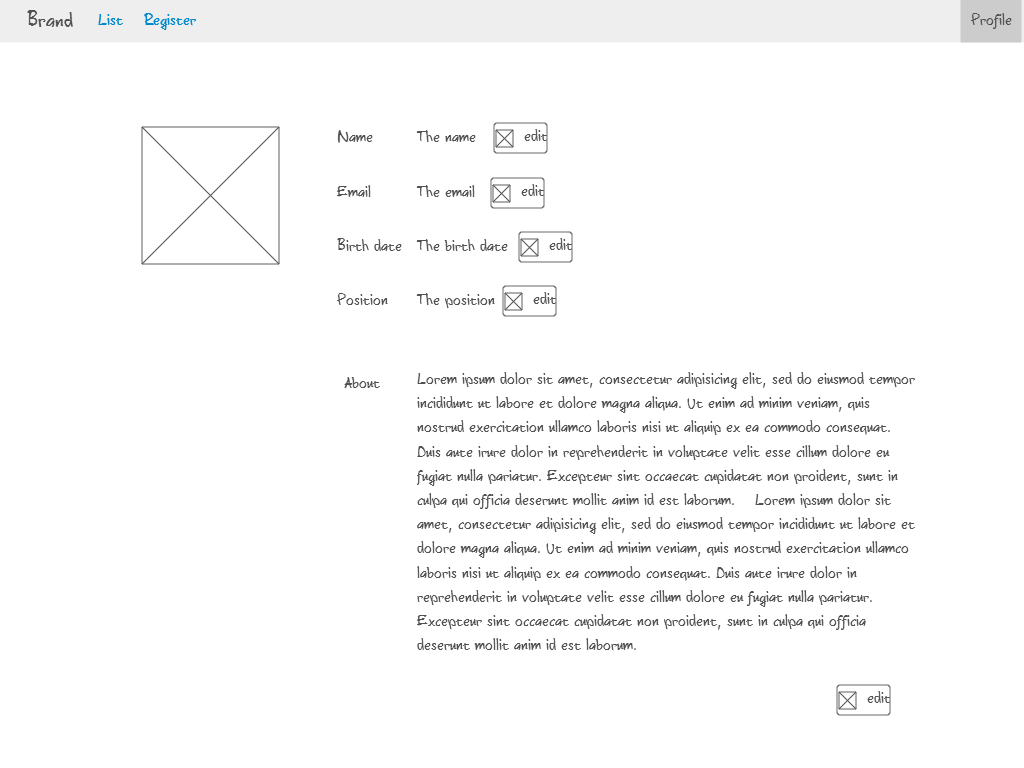
1. sometimes users input incorrect data and realise it only after submission when he cannot change them
2. currently to change user data manual database handling is required; running bare SQL scripts on the production database is hard-to-perform operation which cost
3. some personal data can be seen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

In order to handle these problems this document was created.

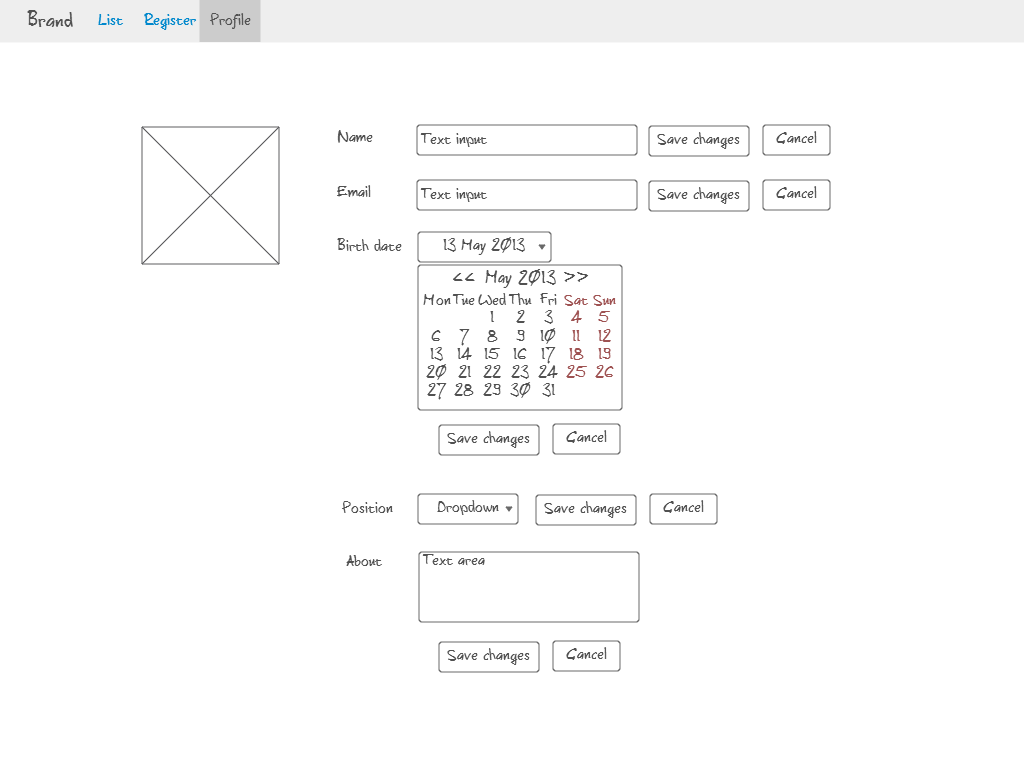
## Vision

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.



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.



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, last name and password must be there as well. Currently authenticated user can edit only 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.User interface

# Validation