

Development of web portal for Distributed  
And Network Algorithmics (DANA) lab  
Ανάπτυξη portal για το εργαστήριο  
Distributed And Network Algorithmics  
(DANA lab)

by Neilos - Christos Psathas

Department of Electrical and Computer Engineering  
University of Thessaly



Volos, 2018

## **ABSTRACT**

The development of a web portal with user-roles is mainly to support a company's web site and help the communication between employees. In our case this web portal is about an academic lab. The purpose of this thesis is the development of a web portal for Distributed And Network Algorithmics (DANA) lab. The supervisor of it is Dimitrios Katsaros. The web portal will present the academic profiles of DANA lab members.

## **ΠΕΡΙΛΗΨΗ**

Η ανάπτυξη ενός portal με ρόλους, που δύναται να υποστηρίξει εταιρικό site για προβολή των δραστηριοτήτων της εταιρείας και εσωτερική επικοινωνία των μελών της. Η εφαρμογή που κάνουμε είναι για την περίπτωση ενός πανεπιστημιακού εργαστηρίου. Ο σκοπός αυτής της διπλωματικής είναι η ανάπτυξη διαδικτυακής εφαρμογής για το εργαστήριο Distributed And Network Algorithmics (DANA lab). Υπεύθυνος του εργαστηρίου είναι ο κύριος Δημήτριος Κατσαρός. Αυτή η διαδικτυακή εφαρμογή θα παρουσιάζει τα ακαδημαϊκά προφίλ των μελών του εργαστηρίου.

## **ACKNOWLEDGEMENTS**

Με την εκπόνηση της διπλωματικής μου κατ' αρχήν όλους τους καθηγητές μου για όλα αυτά που έκαναν (ή δεν έκαναν) και για όλα αυτά που μου έμαθαν όλα αυτά τα χρόνια, με ποιο σημαντικό τον κύριο Κατσαρό ως επιβλέπον για την διπλωματική μου αλλά και ως καθηγητή μου σε 6 μαθήματα. Ακόμη, θέλω να ευχαριστήσω τους κοντινούς μου φίλους καθώς και την οικογένεια για τη συνεχή στήριξη και τη βοήθεια. Πάνω από όλα όμως, θέλω να πω δύο μεγάλα ευχαριστώ, πρώτον στον αδερφό μου ο οποίος ήταν ο “υπεύθυνος” που διάλεξα αυτή τη σχολή και επίσης στον παππού μου όπου ήταν κυριολεκτικά πάντα δίπλα μου!

<b>ABSTRACT</b>	<b>1</b>
<b>ΠΕΡΙΛΗΨΗ</b>	<b>1</b>
<b>ACKNOWLEDGEMENTS</b>	<b>2</b>
<b>Introduction</b>	<b>4</b>
<b>Functionality</b>	<b>4</b>
Login / Logout	4
User's page	4
Edit informations	4
<b>User Interface (UI)</b>	<b>5</b>
Desktop / Tablet UI	5
Mobile UI	9
<b>The Ruby on Rails framework</b>	<b>15</b>
Overview	15
MVC	15
Database	15
Scheme	15
Initial data	16
Front-end assets and libraries used	16
jQuery	16
jQuery plugins	16
Bootstrap	17
Font-awesome	17
The MVC in Rails	17
Models	17
Controllers	17
Views	17
<b>Future additions</b>	<b>18</b>
<b>References</b>	<b>19</b>

## Introduction

Is very important for academic professors and diploma students to have a well presented academic profile web page. In the following sections we will describe the user interface of the DANA Lab Web Portal and then the framework along with the code and libraries that used to build it. The last section will present some ideas of future additions/improvements.

## Functionality

The DANA web portal have these main features:

- Login / Logout of registered user
- Presentation of his/her informations
- Registration of new users
- Deletion of registered users

All of the above should be done with easy and user-friendly way without assumptions for the technical knowledge of the user. Each user have the ability to edit his/her page. The registration and deletion of users is done by a special type of users (called admins) that have the “*admin rights*”. Admin users have also the ability to edit the informations of other users, promote normal users to admins and see the list of registered users.

## Login / Logout

The registered users have the ability to login to the portal by providing the unique username and password of their accounts. The login page is the homepage of the app. When a user is logged in, is then redirected to his/her personal profile. A logged user may logout of the app at any time, so he/she then can login again with a different account.

## User’s page

This is the main page of the app. In this page is presented the informations of a user. This includes the bio, the education training, the academic positions, the honors and awards, the teaching activities, the publications and the research projects of the user.

## Edit informations

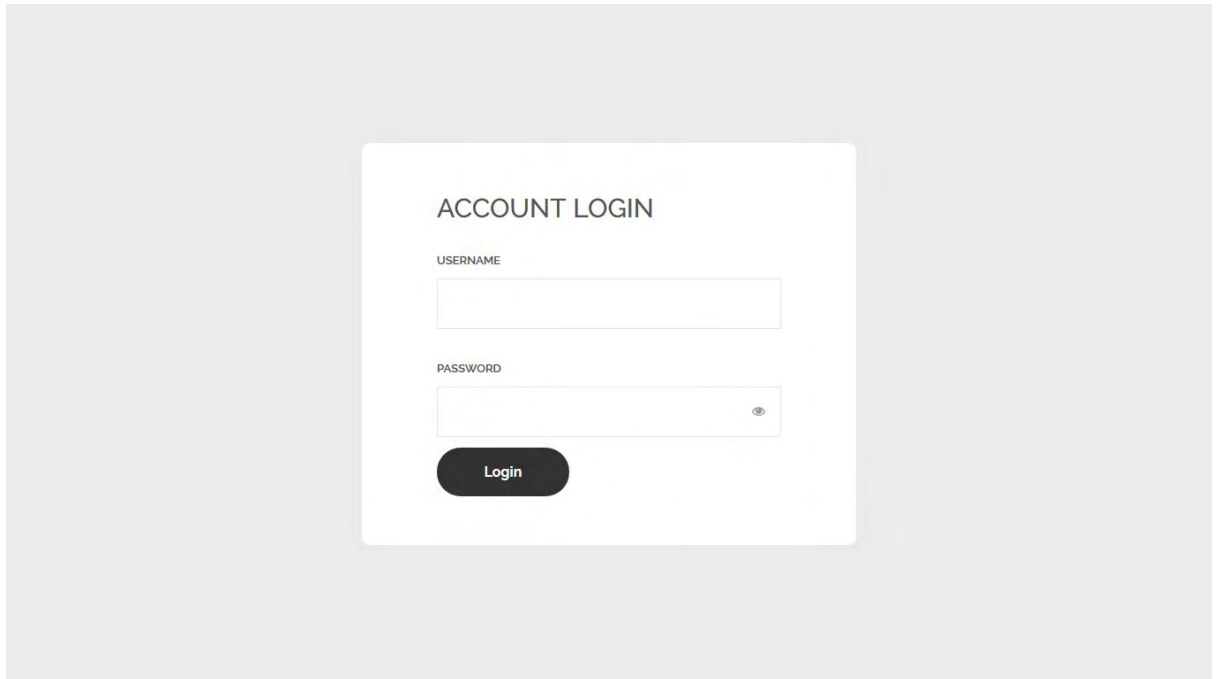
When a user is logged, have the ability to alter the informations of his/her personal profile.

## User Interface (UI)

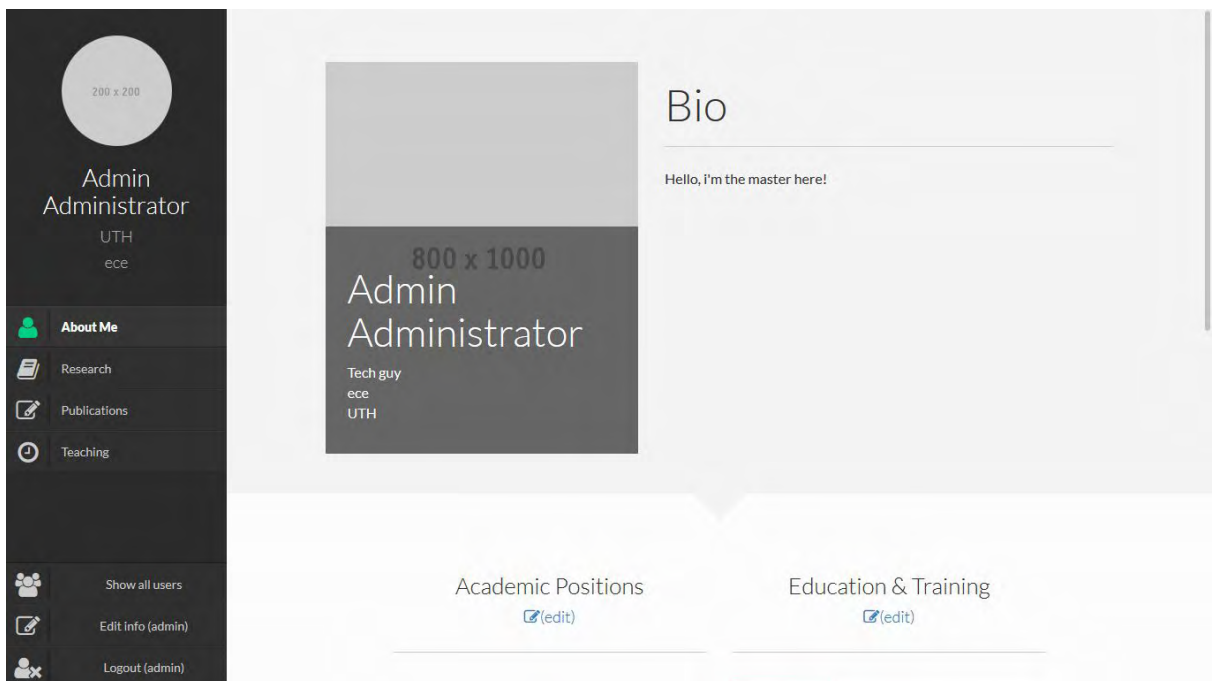
Below there is screenshots of the web app. The UI is responsive, it means that the content is presented different in mobile devices with narrow screens, utilizing the vertical scrolling and the touch functionality, it also resizing the fonts and images to make them easier to read and fit to the screen.

### Desktop / Tablet UI

#### 1. Login page



#### 2. User's page



200 x 200

**Admin Administrator**  
UTH  
ece

**About Me**

- Research
- Publications
- Teaching

Show all users  
Edit info (admin)  
Logout (admin)

Academic Positions [\(edit\)](#)

Education & Training [\(edit\)](#)

**Ph.D. 1983**  
Admin degree  
UTH.

Honors, Awards and Grants [\(edit\)](#)

200 x 200

**Admin Administrator**  
UTH  
ece

**About Me**

- Research
- Publications
- Teaching

Show all users  
Edit info (admin)  
Logout (admin)

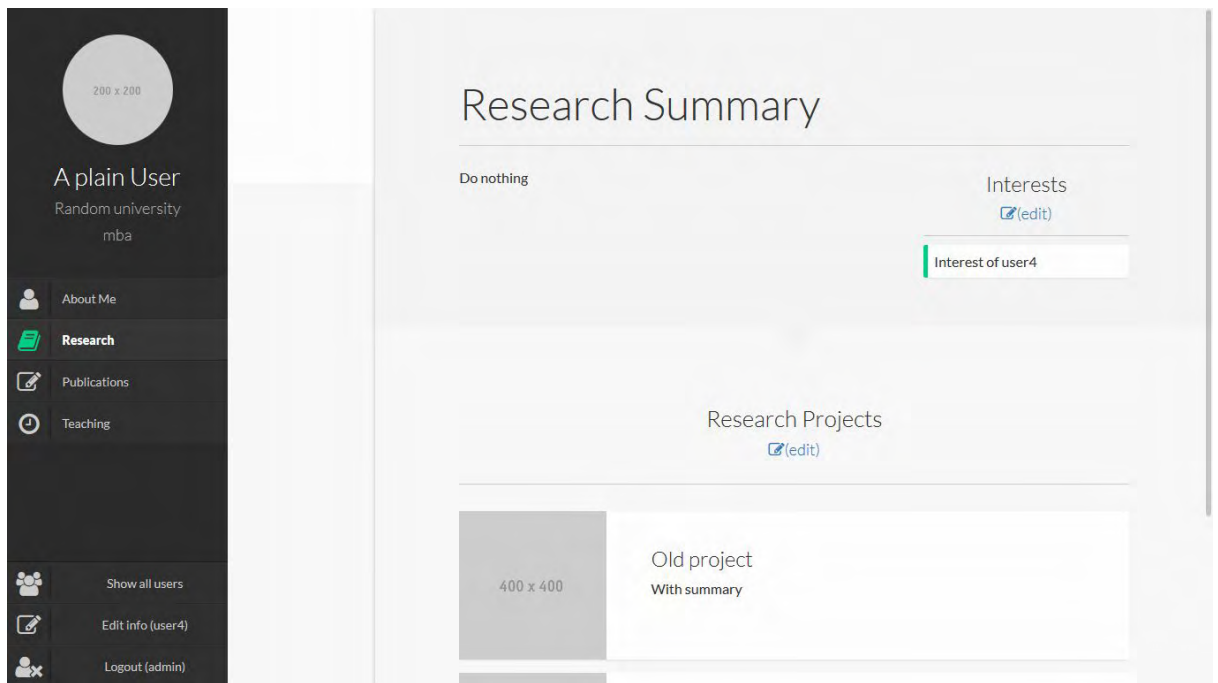
Admin degree  
UTH.

Honors, Awards and Grants [\(edit\)](#)

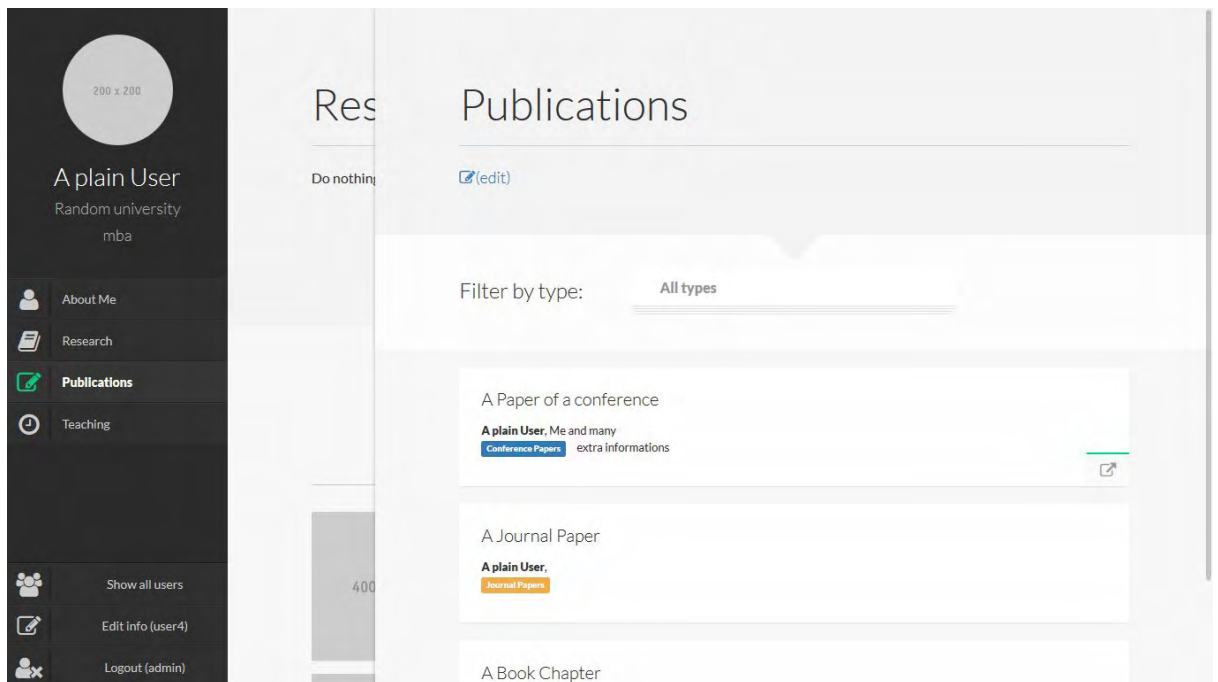
Other Activities [\(edit\)](#)

Surfing on superuser

### 3. Research section

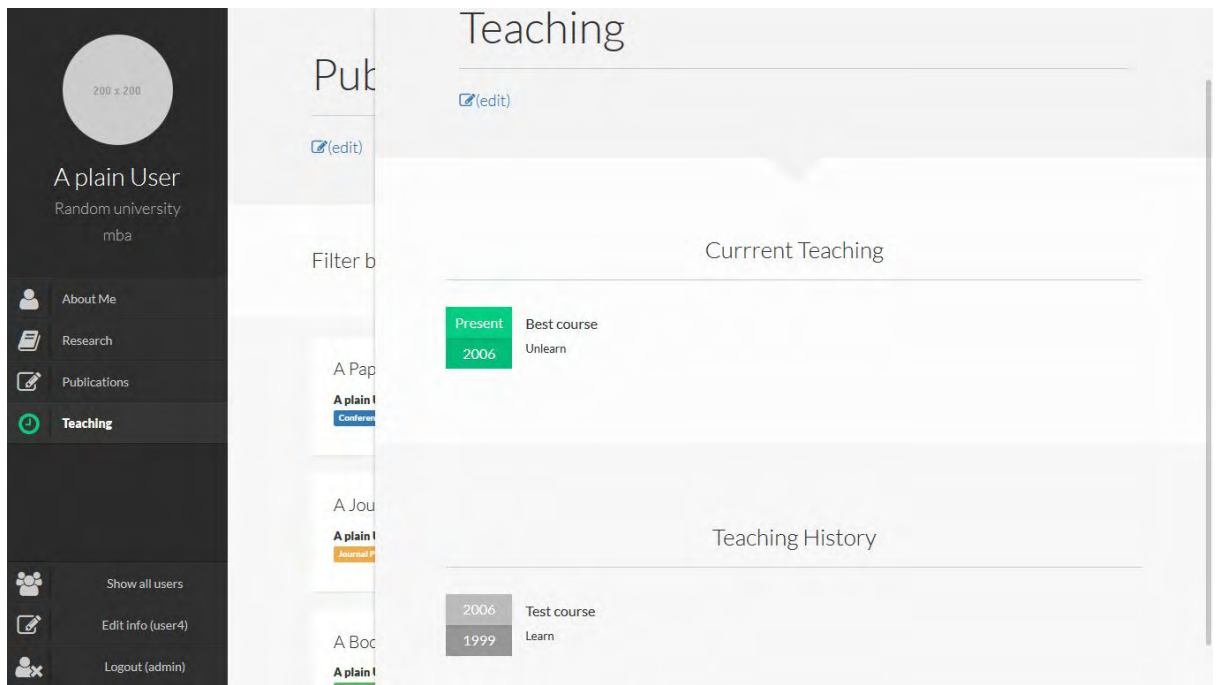


### 4. Publications section





5. Teaching section



6. All users page

**Users**

Is admin	Username	Email	Password	First name	Last name	Academic title	Department	University	Bio	Telephone	Address	Research summary
true	admin	dana@dana.uth.gr	1234	Admin	Administrator	Tech guy	ece	UTH	Hello, i'm the master here!	24210123456	Earth	Doing everything <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
false	user4	userFour@uth.gr	4444	A plain	User	undergraduated student	mba	Random university	Hi, guys	0987654321	Town	Do nothing <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
true	nilos	npsathas@uth.gr	rooty	Nilos	Psathas	postgraduated student	inf	University of Thessaly	This is me	6978660917	Statha 8, Volos, Greece	Many things <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[New User](#)

## 7. Edit informations page

### Editing User

Is admin

Username

Email

Password

First name

Last name

Academic title

Department

University

Bio

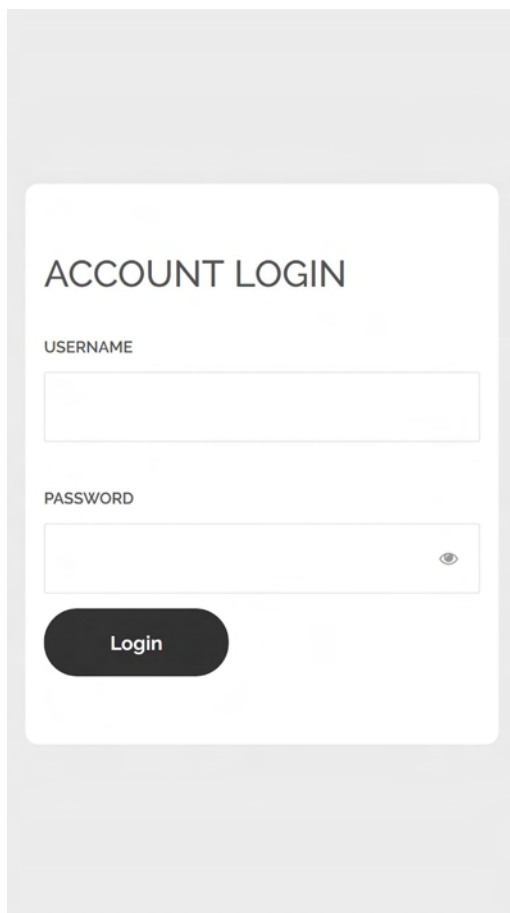
Telephone

Address

Research summary

## Mobile UI

### 1. Login page

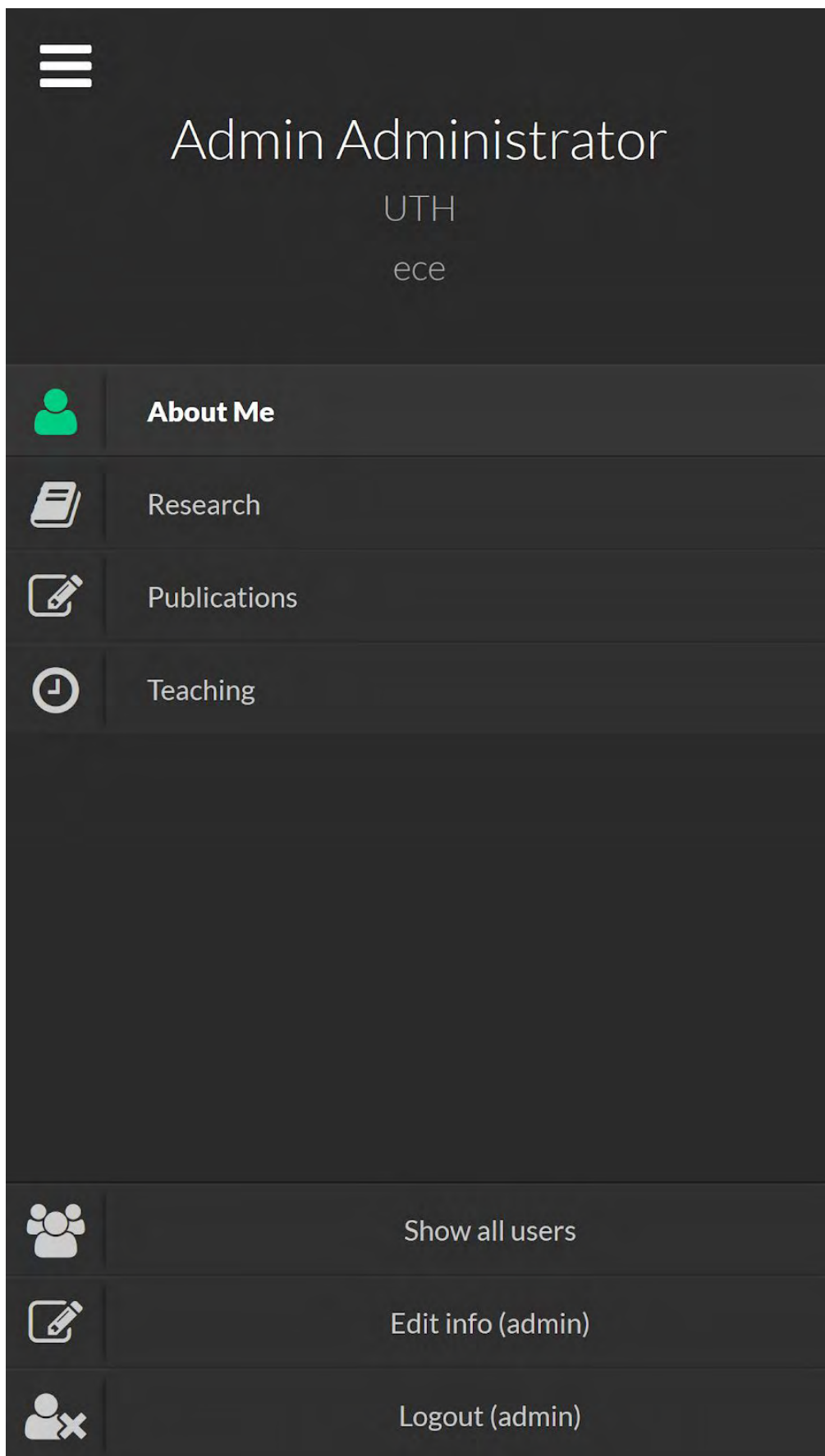


ACCOUNT LOGIN

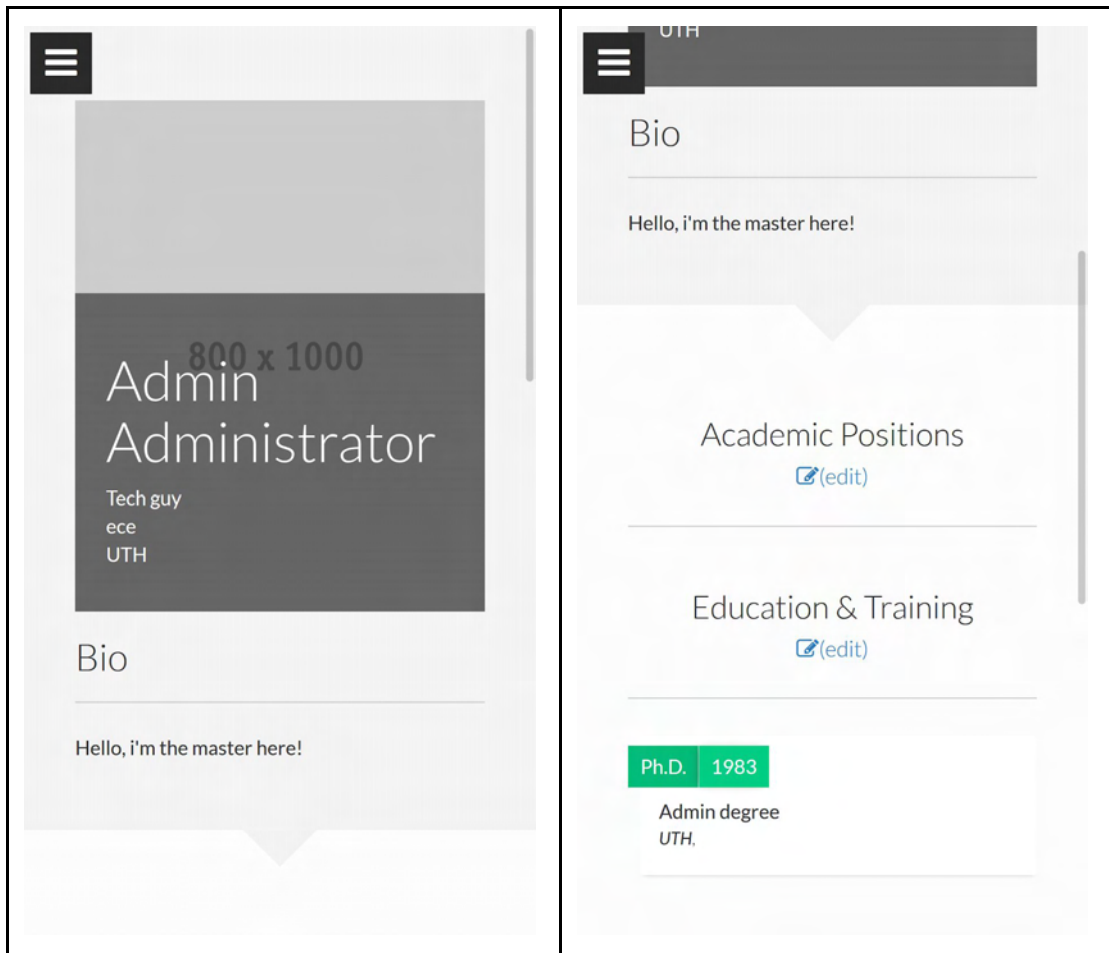
USERNAME

PASSWORD

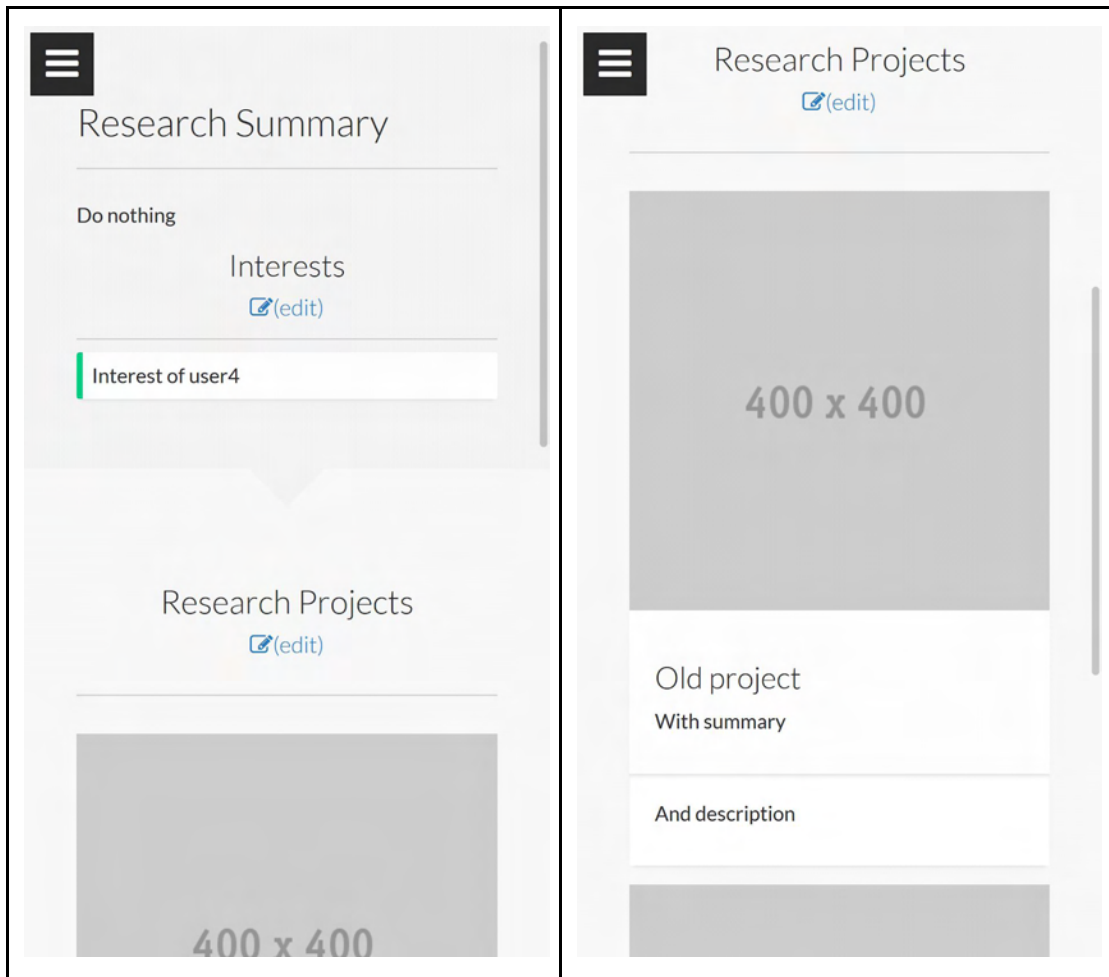
## 2. Menu



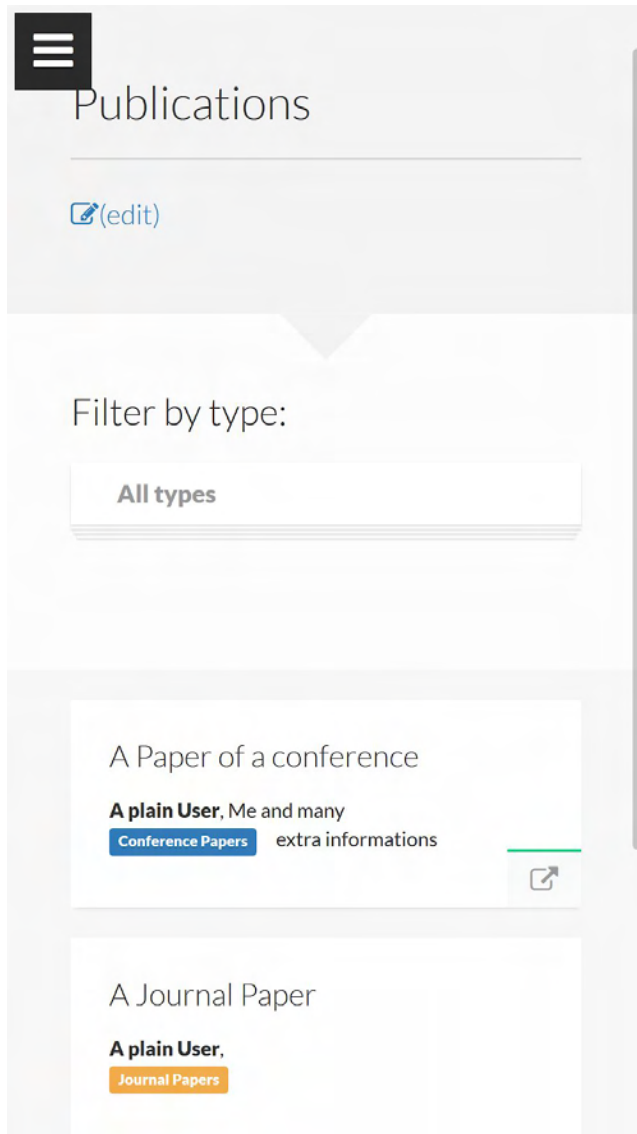
### 3. User's page



#### 4. Research section



## 5. Publications section



## 6. Teaching section

Teaching

(edit)

### Current Teaching

Present	Best course
2006	Unlearn

### Teaching History

2006	Test course
1999	Learn

# The Ruby on Rails framework

## Overview

Ruby on Rails, or Rails, is a server-side web application framework written in Ruby under the MIT License. Rails is a model–view–controller (MVC) framework, providing default structures for a database, a web service, and web pages. It encourages and facilitates the use of web standards such as JSON or XML for data transfer, and HTML, CSS and JavaScript for display and user interfacing. In addition to MVC, Rails emphasizes the use of other well-known software engineering patterns and paradigms, including convention over configuration (CoC), don't repeat yourself (DRY), and the active record pattern.

## MVC

Model–view–controller (MVC) is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development.

## Database

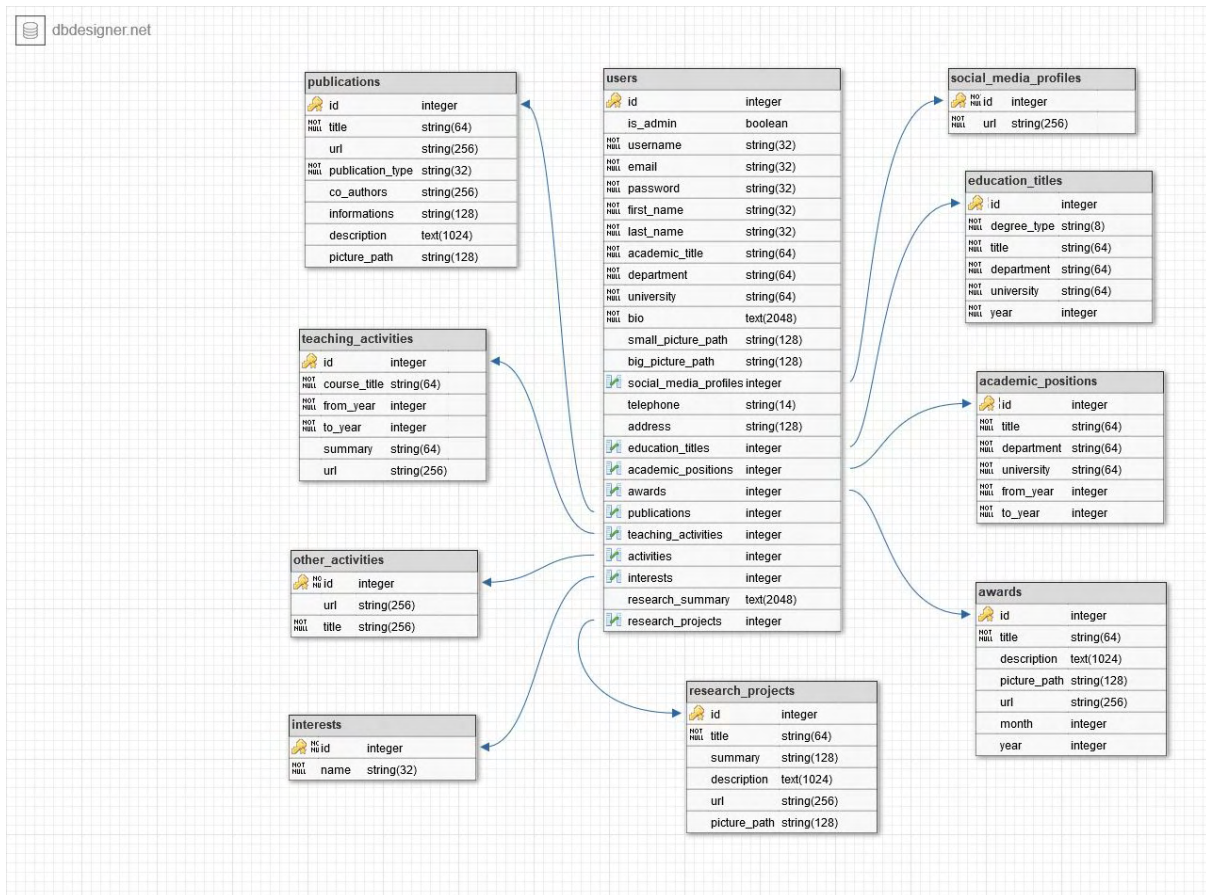
All the dynamic data of the app are stored in a database. For the DANA Lab Web Portal we chose to use the SQLite database. The SQLite is a server-less relation database with small memory footprint and cpu demands. It is one of the best choices for applications with little amount of data. All the data are stored in a single file that is loaded to the main memory upon application's startup.

The Rails framework have its own API for reading and writing from the database. The developer is using the API, which is database agnostic and is simple object manipulation.

## Scheme

The scheme of the database is also managed by the Rails framework. The developer runs some special commands and the framework generates the SQL code to create the database tables and the relations between them. Our application's scheme is this:





## Initial data

Because the registration of new users is done only from the admins, the application should have a pre-registered user with “admin rights”. The framework also has the feature of having some saved data before the application run for the first time. This is done in a special file called seeds.rb

## Front-end assets and libraries used

### jQuery

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications.

### jQuery plugins

jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation,

advanced effects and high-level, themeable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

## Bootstrap

Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only. Bootstrap is the second most-starred project on GitHub, with more than 121,000 stars.

## Font-awesome

Font Awesome is a full suite of 628 pictographic icons for easy scalable vector graphics on websites.

## The MVC in Rails

### Models

Active Record is the M in MVC - the model - which is the layer of the system responsible for representing business data and logic. Active Record facilitates the creation and use of business objects whose data requires persistent storage to a database. It is an implementation of the Active Record pattern which itself is a description of an Object Relational Mapping system. Active Model is a library containing various modules used in developing classes that need some features present on Active Record.

### Controllers

Action Controller is the C in MVC. After the router has determined which controller to use for a request, the controller is responsible for making sense of the request and producing the appropriate output. Luckily, Action Controller does most of the groundwork for you and uses smart conventions to make this as straightforward as possible. For most conventional RESTful applications, the controller will receive the request (this is invisible to you as the developer), fetch or save data from a model and use a view to create HTML output. A controller can thus be thought of as a middleman between models and views. It makes the model data available to the view so it can display that data to the user, and it saves or updates user data to the model.

### Views

Interaction between Controller and Model in the Model-View-Controller triangle starts and ends in View. The Controller is responsible for orchestrating the whole process of handling a request in Rails, though it normally hands off any heavy code to the Model. But then, when it's time to send a response back to the user, the Controller hands things off to the View. In

broad strokes, this involves deciding what should be sent as the response and calling an appropriate method to create that response. If the response is a full-blown view, Rails also does some extra work to wrap the view in a layout and possibly to pull in partial views.

## **Future additions**

The application is in an early stage and there are a lot of features that may be added, some of them are:

- Description of fields and data validation on “edit page”
- Invitation of new users to portal through email
- Forget password mechanism
- Microblogging
- In-app messages (and live chat maybe)
- Social network features - follow users to take notifications

## References

1. [ruby-doc.org](http://ruby-doc.org)
2. [Ruby on Rails Guides](http://guides.rubyonrails.org)
3. [Bootstrap documentation](http://getbootstrap.com)
4. [jQuery API](http://jquery.com)
5. [MVC wikipedia page](http://en.wikipedia.org/wiki/Model-view-controller)
6. [Font awesome](http://fontawesome.com)
7. [Codecademy.com](http://codecademy.com)