



Introduction to Ruby on Rails

Software Engineering II
WS 2015/16

Thomas Kowark
Prof. Plattner, Dr. Uflacker
Enterprise Platform and Integration Concepts group

Introduction to Ruby on Rails



1. Ruby & Ruby on Rails

- What is Ruby on Rails?
- A few words about Ruby
- Rails' core components
- RESTful architecture

2. Your first Rails application

3. Your introductory Rails exercise

4. Additional Literature

What is Ruby on Rails?

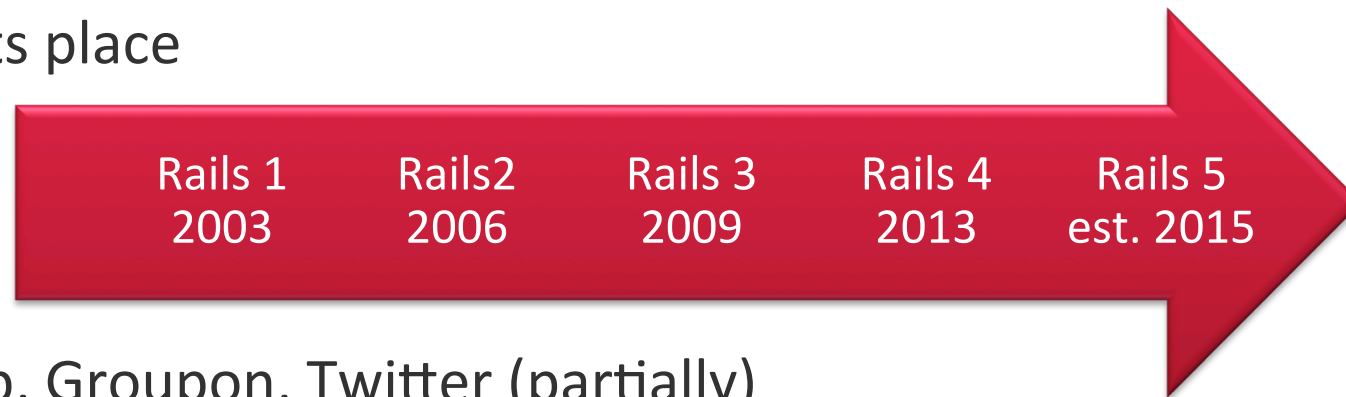


Web application development framework written in Ruby

- <http://rubyonrails.org/>

Philosophy

- "Don't repeat yourself" – DRY
- Convention over Configuration – there is "the Rails way"
- RESTful architecture
- Everything in its place



- Used by Github, Groupon, Twitter (partially)

A few words about Ruby

<http://www.ruby-lang.org/>

- Dynamic, reflective, general-purpose, object-oriented
- Influenced by Perl, Smalltalk, Eiffel, and Lisp
- Open-source, mature software
- Matz's Ruby Interpreter (MRI) versions:

Ruby 1.0
1996

Ruby 1.8.7
2010

Ruby 1.9.3
2011

Ruby 2.0.0
2013

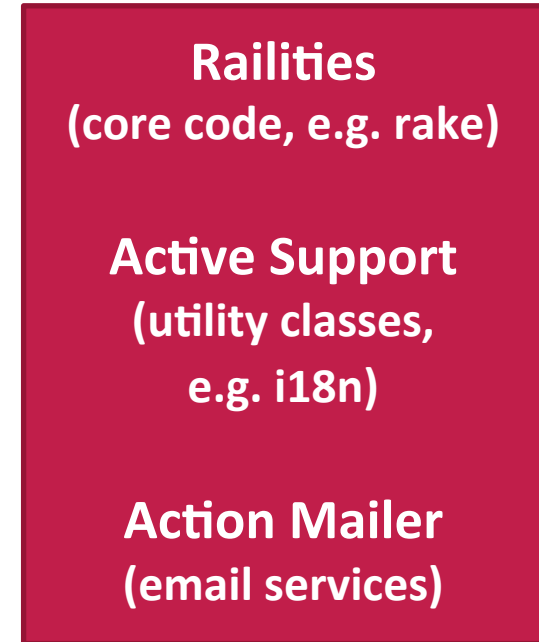
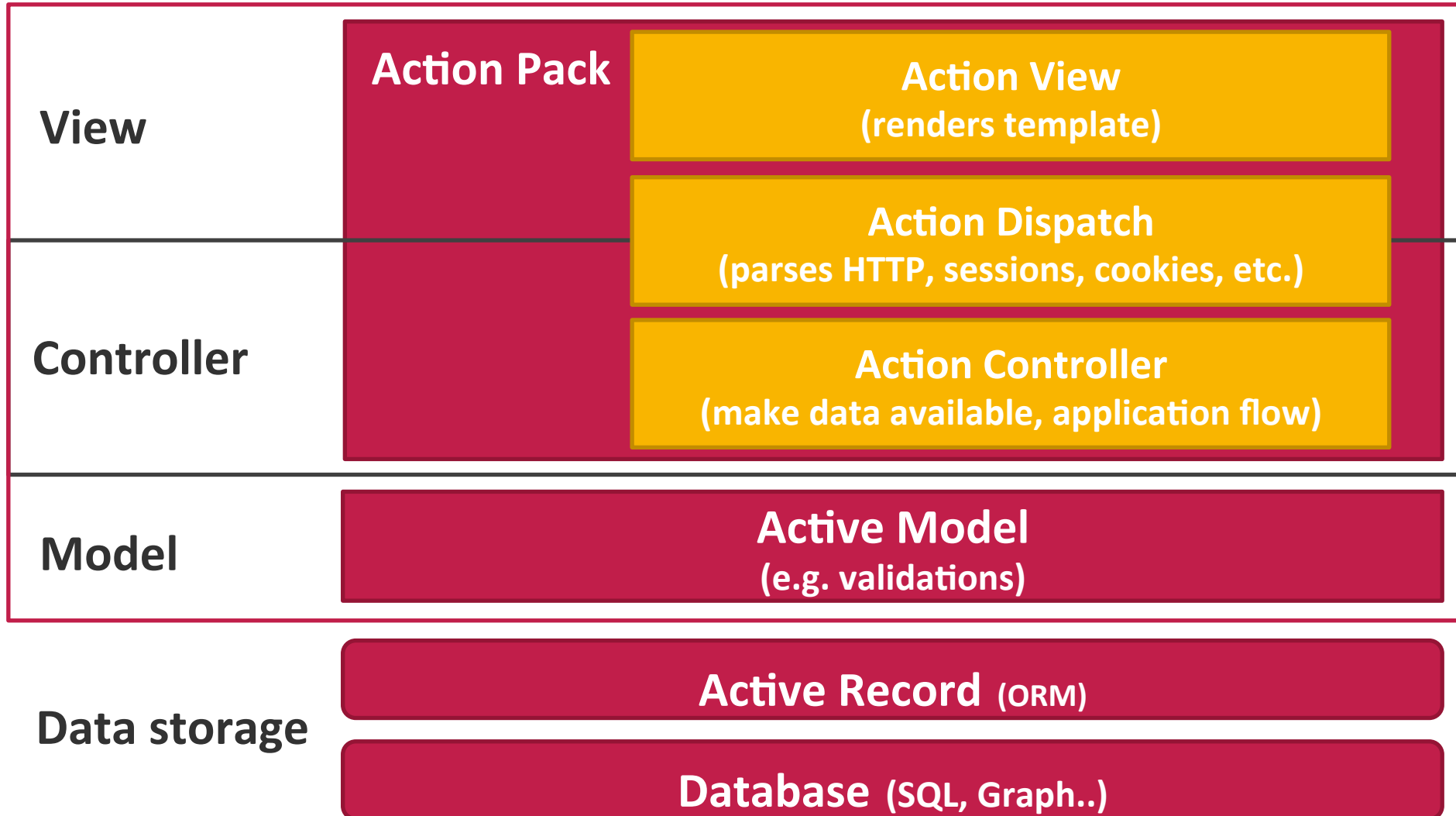
Ruby 2.2.2
2015

- Additionally different VMs available (JRuby, Rubinius, IronRuby, Maglev)

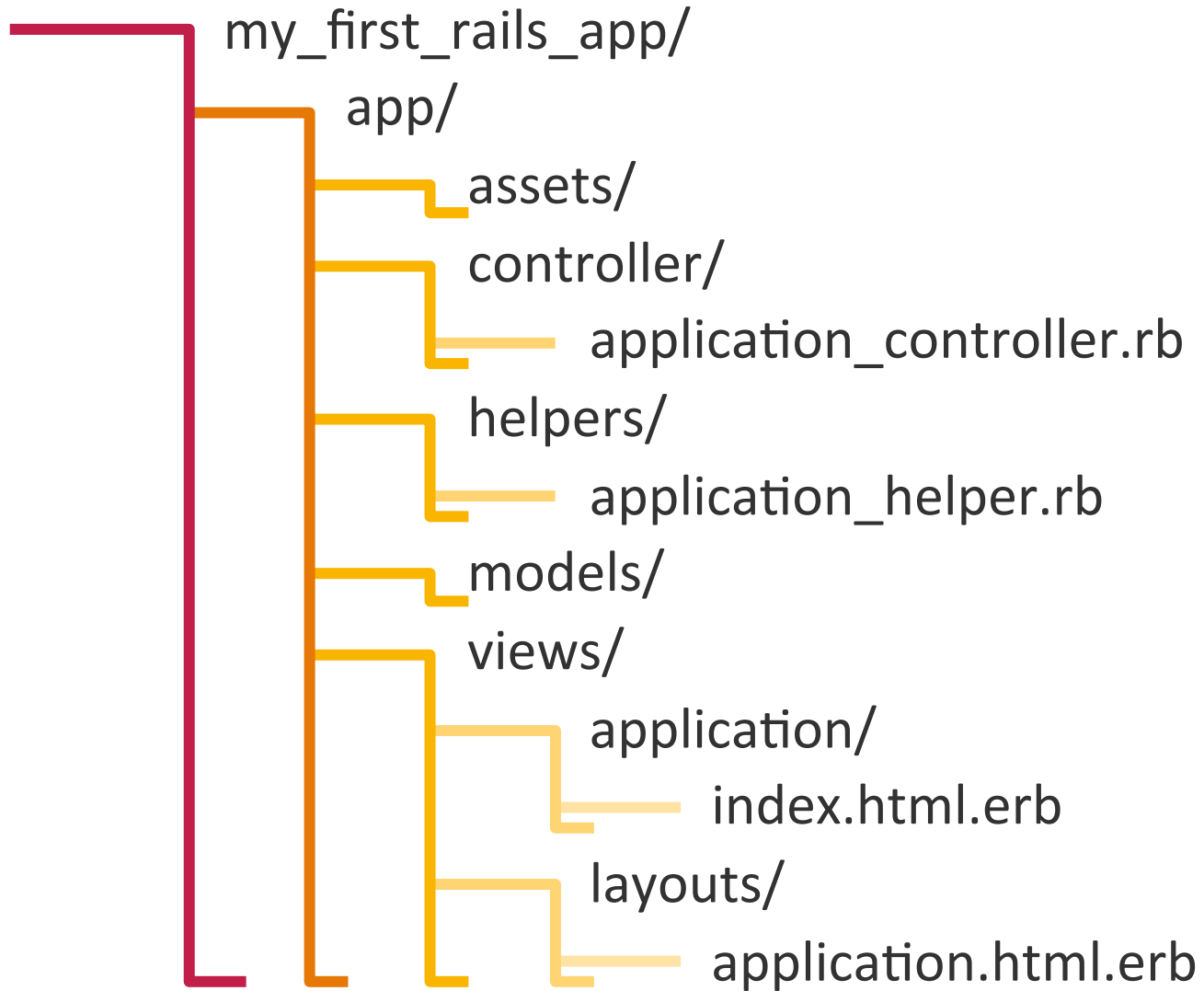


Yukihiro "Matz" Matsumoto with R. Stallman

Rails Core Components



Rails Application Layout



RESTful Architecture



- **Representational State Transfer (REST)** is a software architecture style for distributed systems
- **Principles**
 - Uniform Interface
 - Stateless Interactions
 - Cacheable
 - Clients and servers
 - Layered System
 - Code on Demand (optional)
- **Largest RESTful implementation: World Wide Web**

RESTful Architecture - HTTP verbs



- REST supports all 4 HTTP 1.1 verbs: GET, PUT, POST, DELETE
- Differentiation of collections and individual elements

Resource	GET	PUT	POST	DELETE
Single element http://localhost:3000/authors/1	Retrieve	Update or create	Create	Delete
Collection http://localhost:3000/authors	List	Replace	Create	Delete

Examples of Routes



- GET / # invoke “home” controller
- GET /authors # retrieve a list of all authors
- GET /authors/new # get the form to enter a new author
- POST /authors # create a new author
- GET /authors/1 # show details of the first author
- GET /authors/1/edit # get the form to edit the first author
- PUT /authors/1 # update the first author
- DELETE /authors/1 # delete the first author

Introduction to Ruby on Rails



1. Ruby & Ruby on Rails
- 2. Your first Rails application**
3. Your introductory Rails exercise
4. Additional Literature

How to Start?



- **Option 1:** You use *Mac* or *Linux*
 - Install and use Ruby on Rails directly on your OS
 - Ruby version manager (e.g. RVM, rbenv) if older versions of Ruby should be kept
 - http://guides.rubyonrails.org/getting_started.html#installing-rails
 - Or use option 2
- **Option 2:** You have *Windows* or want to use a VM (**recommended**)
 - We prepared one for you via Vagrant (<https://www.vagrantup.com/>)
 - Uses VirtualBox in the backend (free on all platforms) (<https://www.virtualbox.org/>)
 - Use your own tools & editors, run the project in a headless VM
 - See project README for setup instructions
- **Option 3:** You have *Windows* and install Ruby on Rails directly on your OS
 - Tends to consume some time, might cause problems with certain dependencies
 - <http://railsinstaller.org/en>

Comprehensive RoR tutorial

Recommended to work through / read this hands-on tutorial. Seriously.

http://guides.rubyonrails.org/getting_started.html



 **RailsGuides**

Getting Started with Rails

This guide covers getting up and running with Ruby on Rails.

After reading this guide, you will know:

- ✔ How to install Rails, create a new Rails application, and connect your application to a database.
- ✔ The general layout of a Rails application.
- ✔ The basic principles of MVC (Model, View, Controller) and RESTful design.
- ✔ How to quickly generate the starting pieces of a Rails application.



Info:

Before you start coding, make sure, the correct versions are installed.

```
$ ruby --version  
$ rails --version
```

The following slides give a general overview

rails - Main executable



Start interactive shell to test out ideas

```
$ rails console
```

Start new rails application

```
$ rails new
```

Generate boilerplate for models, controllers & views

```
$ rails generate
```

Start the development server

```
$ rails server
```

Start a direct database shell

```
$ rails dbconsole
```

- Example: generate model, controller and view without controller specs

```
$ rails g scaffold author last_name:string  
homepage:string --controller-specs false
```

Bundler - Ruby package manager

- Ruby libraries are packaged as "gems"
- Online repository at <https://rubygems.org/>
- Bundler resolves dependencies of gems
- Gemfile holds a list of required gems
 - Specify versions, e.g. `gem 'rails' >= '4.1.6'`
 - Alt. sources, e.g. `:github => "tkowark/sawyer"`
- Gemfile.lock is populated with resolved dependencies
 - Should be under version control

Manually install a gem (Ruby package)

```
$ gem install
```

Install all gems listed as dependencies in Gemfile

```
$ bundle install
```

```
# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '4.1.6'
# Use sqlite3 as the database for Active Record
gem 'sqlite3', group: :development

# use postgresql in production (for deployment on heroku)
gem 'pg', group: :production

# Use Bootstrap, see app/assets/stylesheets
gem 'twitter-bootstrap-rails'
# Use SCSS for stylesheets
# gem 'sass-rails', '~> 4.0.3'
# Use Uglifier as compressor for JavaScript assets
gem 'uglifier', '>= 1.3.0'
# Use CoffeeScript for .js.coffee assets and views
# gem 'coffee-rails', '~> 4.0.0'
# See https://github.com/sstephenson/execjs#readme
gem 'therubyracer', platforms: :ruby
```

Info:

Gemfile.lock contains all the actually installed versions of gems.

rake - Ruby make



List all available rake commands

```
$ rake -T
```

List all configured routes

```
$ rake routes
```

Setup the database and run all migrations

```
$ rake db:setup && rake db:migrate
```

Replace database with db layout from db/schema.rb
Do not run migrations.

```
$ rake db:schema:load
```

Run Rspec (testing framework for RoR) tests

```
$ rake spec
```



Info:

Running `schema:load` is advisable when setting up a completely new project. It is not intended to work around bad migrations.

Git – distributed version control system



■ Install Git:

- sudo apt-get install git
- <http://git-scm.com/> (Installers for all systems)

■ Setting up user name and email:

- Mandatory to commit changes
- Use your github credentials!

```
$ git config --global user.email "vorname.nachname@student.hpi.de"  
$ git config --global user.name "Max Mustermann"
```

■ Alternative: setting parameters only for one project:

```
$ cd /path/to/your/project  
$ git config user.email "vorname.nachname@student.hpi.de"  
$ git config user.name "Max Mustermann"
```


Git workflow - committing a change



Checkout remote repository to local copy

```
$ git clone https://github.com/hpi-swt2/wimi-portal
```

Change main layout template `app/views/layouts/application.html.erb`

Stage changes (add files from working copy to repository index)

```
$ git add app/views/layouts
```

List changes to be committed

```
$ git status
```

Commit with commit messages. Reference Github issue #25

```
$ git commit -m "Fixed issue #25"
```

Fetch and merge changes from remote repository

```
$ git pull
```

Publish local commits

```
$ git push
```

Introduction to Ruby on Rails



1. Ruby & Ruby on Rails
2. Your first Rails application
- 3. Your introductory Rails exercise**
4. Additional Literature

Exercise 1 - Code School



- Goals
 - Get familiar with Ruby on Rails
 - Create necessary accounts for the project
- Tasks
 - Create a Github account
 - Link it with CodeSchool.com (unless you already have an account)
 - Complete Rails coding school: <http://railsforzombies.com>
 - Complete Git Tutorial: <https://www.codeschool.com/courses/try-git>
 - When done, create a screenshot of your report card
- Deadline: Nov 6, 11:59 pm CET (firm)

Exercise 2 - Develop a Rails App



■ Goals

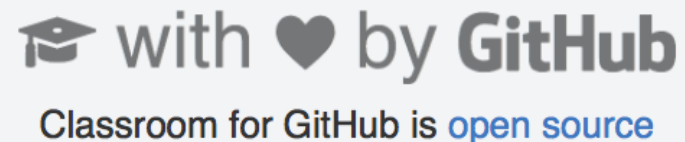
- Deepen your understanding of Rails Apps
- Get used to TDD/BDD

■ Tasks

- Sign up on Github Classroom
 - <https://classroom.github.com/assignment-invitations/ebc02d7d20638a7ef2660434c04136c2>
 - Repository might take some time to receive initial skeleton code (be patient...)
- Make the tests “green”
 - Start with ‘rake spec’
 - Run single feature tests with ‘rspec spec/features/.../..._spec.rb’
- Commit your screenshot from Exercise 1

■ Deadline

- Nov 6, 11:59 pm CET (firm)
- POs are exempt from this task.

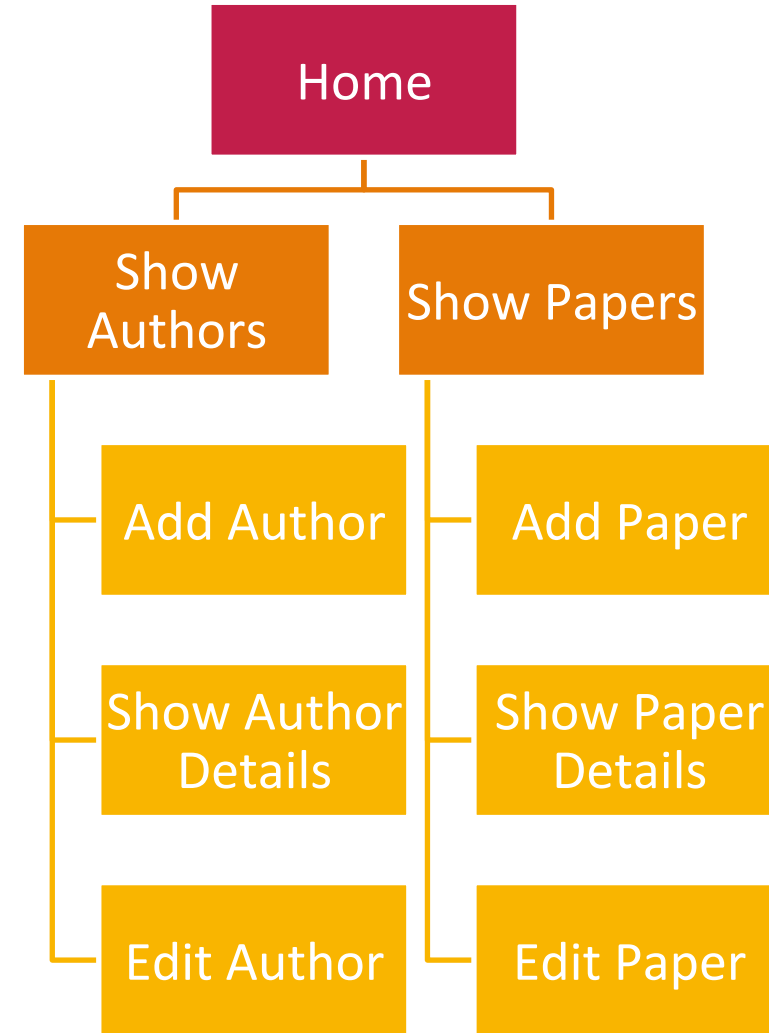
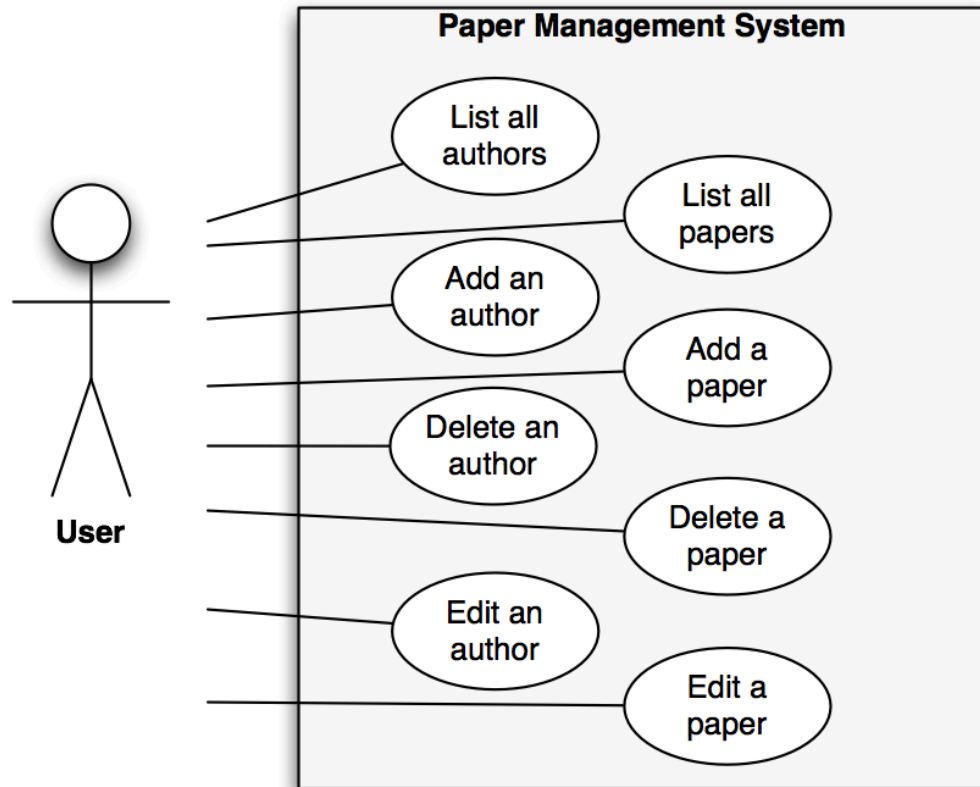


Exercise 2 - Requirements



- Web-based paper management system
 - Author
 - First name
 - Last name
 - Homepage
 - Paper
 - Title
 - Publication
 - Year
 - A paper has many authors (max. 5)
 - An author has many papers
- We prepared a Rails application for you

Exercise 2 - Use Cases & Site Map



Exercise 2 - Mock Up

<https://gomockingbird.com/mockingbird/index.html?project=v890g6l>

Paper Management System

[Show papers](#)
[Show authors](#)

Listing authors

Name	Homepage	
Hasso Plattner	http://www.hpi.uni-potsdam.de/personen/stifter.html	Edit Delete

[Add author](#) | [Back](#)

First name: Hasso

Last name: Plattner

Homepage: <http://www.hpi.uni-potsdam.de/personen/stifter.html>

Papers:

[This is a paper](#)

[Edit](#) | [Back](#)

Editing author

First name

Last name

Homepage

[Show](#) | [Back](#)

Additional Literature



General literature

- Ruby, S.; Thomas, D.; Hansson D. H.: Agile Web Development with Rails 4, 2013 (ebook)
- Swicegood, T.: Pragmatic Guide to Git (P3.0), 2012 (ebook)
- Rappin, N.: Rails Test Prescriptions, 2012
- Rasmusson, J.: The Agile Samurai (P4.0), 2012 (ebook)
- Pichler, R.: Agile Product Management with Scrum, 2010
- Cohn, M.: Succeeding with Agile, 2012
- Larman, C.; Vodde, B.: Scaling Lean & Agile Development, 2009
- Ludewig, J.; Lichter, H.: Software Engineering, 2006
- Sommerville, I.: Software Engineering, 2004

Ebooks will be made available for chosen books

Next Weeks' Schedule



Week 1 (Oct 12 – Oct 16)

- Introduction lectures

Week 2 (Oct 19 – Oct 23)

- Find teams, **enroll!**
- Code School exercise
- 11-12:30 Lecture on Scrum
- 13:30 – 15 Scrum Lego Exercise

Week 3 (Oct 26 – Oct 30)

- POs: Customer meeting
- Ruby on Rails exercise
- Lecture on Git and testing

Week 4 (Nov 2 – Nov 6)

- Kick-off presentation
- Start of project

Introduction to Ruby on Rails



1. Ruby & Ruby on Rails
 - What is Ruby on Rails?
 - Rails' core components
 - RESTful architecture
2. Your first Rails application
 - Folder structure
 - rails, rake, git
3. Your introductory Rails exercise
 - Code School
 - Github Classroom
4. Additional Literature