Adv. Server-Side : Project 2

Prepared for: Oonagh O Brien

Prepared by: Robert Gabriel , Aron Natnael, Piotr Kawalec

10th November 2014

EXECUTIVE SUMMARY

Project Specification

20% will be given for a report and presentation by each group which evaluates the advantages and disadvantages or three php based application frameworks - one to be Symfony. You will need to identify at least 8 criteria to evaluate each framework under and each criteria is given a weighting, the weighting of the criteria must be explained. An evaluation report must be produced, which must include a table with scores for each framework, under each criteria and details on the evaluation and the findings. This work should be completed by the 10th of November and each group will be present their findings in the lab that week. As part of their presentation each group will identify the application framework they will use for the remainder of the assignment work and outline the web server application they will develop.
Deliverables :- Report on analysis of frameworks, specification of web server application to be developed in phase 2, group presentation.

Frameworks

Coding without a framework works well only for small projects.
The more code the bigger chance to get ‘spaghetti code’ and file disorganisation. That leads to difficulties with code-reuse and setup. Also, making a change in one place can easily break something in another. Testing the code is difficult and all tasks impose challenges to team members.

With framework you save on unnecessary work and which could require redundant changes. The code is clean and easy to reuse. Organisation provides easy navigation and changes in some part of the code are reflected in all places. Frameworks include pre-built and pre-tested tools which make development quicker and easier (forms, security, error reporting).

JUDGING AREA

In the next few pages you can see the comments and evaluation on areas that we decided to test, for each of our frameworks picked. We decided the best way to display our results was through bullet points.

The frameworks picked where Symfony 2 ,Cakephp and Codeigniter.

The areas include:

1. Learning curve
2. Documentation
3. Support
4. Flexibility
5. Community
6. Performance
7. Libraries
8. Special Features
9. Projects types
10. Setup

Our final decision, of the framework we picked, will be based on which one is the easiest to learn, provides great performance for user usability point of view and what is simply the best framework over all and not the one that is just for one project type.

LEARNING CURVE

The learning curve is the amount of time it takes a new user to learn the system for the first time. We will judge them based on downloading and using the systems based on our knowledge without looking up guides. It was tested on how long we felt comfortable with the system after about 1 hour of use and building the example application.

11. **Symfony** : Long due to many implemented features, user needs to get familiar with whole structure, concepts and terminology used.

12. **Cakephp** : Is close in design to native PHP, that provides short learning time. Not very strict allowing for small errors that won't break the website.
13.
14. **Codeigniter** : Similar learning curve to Cake php, that provides short learning time.

DOCUMENTATION

The documentation is based on how easy and detailed the systems documentation is laid out and to follow. We tested this by how easy it was to search and find the documentation and how well detailed it was on problems we ran into .

1. **Symfony :** Wide and detailed on the official website , provides information to any aspect the framework of functionality from beginners to experts. The layout isn't great nor is the searching.

2. **Cakephp** : Official documentation is quite messy, links are confusing. Description to the topic is concise but unfortunately lacks of details.

3. **Codeigniter :** The framework website provide complete documentation of libraries including details of the properties and methods. Community forums and blogs are also available.

SUPPORT

The support is judged on how popular the online community is and how well the official forms and system is maintained. Along with if its open source or not. We tested it by how quickly we found an answer online with any bugs or just normal questions we had.

1. **Symfony :** Symfony is backed by Sensio Labs which provides support for developers in many different forms (documentation, reference, tutorials). It is also widely supported by community. Many online forums and tutorials offer topic specific help for the users.

2. **Cakephp :** There is good community support available but still you will need to maneuver well before getting to right content. There is Cake DC corporation offering training and consultancy but this comes with a cost.

3. **Codeigniter** : Like for Symfony, EllisLab supports and develops CodeIgniter, along with great forums and a huge user base, its amazing.

FLEXIBILITY

The flexibility of the system is all about how well this flexs to a project and the php standards.
This is judged on how strong and flexible the system is when it comes to coding standards and how strict it is.
We tested it by using an existing project and trying to see how we made it fit into the framework.

1. **Symfony :** Less strict from version 2.0. Allows for selection of components to be used and for third party libraries as well. Has strict naming convention.

2. **Cakephp :** Fakephp ollows convention over configuration principle which frees user from excessive interference with config files. This approach imposes strictness over the naming and structure.

3. **Codeigniter :** Very flexible, user decides how to use the functionality. The framework comes with set of folders that provide structure for application startup.

COMMUNITY

This is judged on how strong the online community is and how long it is there with how active it is, similarly to the support . We judged this by how strong the subreddits for each framework was and how big the community is online such as stackoverflow along with google trends and the awards.

1. **Symfony :** It has many users, tutorials and forums. Information is available to many different aspects of framework functionality.

2. **cakephp :** It has many users as well, which provide great source for sharing the information.Wide range of topics are covered in online textual and video tutorials.

3. **Codeigniter :** It has many users, exchange of experience and troubleshooting on many sources.

PERFORMANCE

This is judged on how fast the mvc and the speed of the system are. We also took into account it was able to perform sql commands from a simple database, along with common tasks in php using build in functions.

1. **Symfony :** With some issues but since v2.0 improved, support for better caching.

2. **cakephp :** General opinion describes it as a slow framework. With the version 3 developers improved over the routing, template and helpers generation.

3. **codeigniter** : Fast due to small size and only few core libraries are needed to provide working environment.

LIBRARIES

This is judged on how many libraries does it support. We tested this on the amount of common libraries and number of libraries that are the most popular and does it support them.

1. **Symfony :** Large choice of libraries covering many aspects of commercial development. Allows for third parties bundles to be included.
2. **cakephp** : CakePHP comes with a built-in functions and classes that cover some of the most common features required in web applications.
3. **codeigniter** : Libraries available cover many common uses, also allows for creating own custom classes, login systems, paypal integration etc.

SPECIAL FEATURES

This is based on how many extra features it allows, what IDE these support, the more unique and useful the better in this case. Extra points for supporting open source IDE's. Along with the use of mapping (ORM), CRUD.

1. **Symfony :** Supported in many IDEs: Eclipse, NetBeans, PHPStorm. Uses own rendering template engine: Twig .
2. **cakephp :** Allows for IDE integration i.e. NetBeans for ease of development. Due to strict naming convention a lot of code is created for the user behind the scene, it is referred as a automagical functionality.
3. **codeigniter :** Easily implements in many different IDEs: PhpDesigner, KomodoEdit, NEtBeans.

PROJECTS TYPES

This is judged on which project types it is suited for. Be that large scale projects, small or medium projects and how well it performs for those projects. The way we tested this was by searching forums and reviews of the frameworks to get the feedback from developers.

1. **Symfony :** Symfony is used for larger projects which is great but mostly for companies.

2. **Cakephp :** Is great for small and medium projects but not so great for larger projects.

3. **Codeigniter :** Is great for all size projects over , as it is adaptable . You have to get more involved with creating the functionally code.

SETUP

This is judged on how easy the system is to set up, How easy was it for us to see the welcome screen . without any errors.

1. **Symfony :** A bit complicated, needs to use additional components: Composer. The folder structure has to be individually created. For this purpose some level of help is provided with the use of command line tool: app/console.

2. **Cakephp :** Easy setup mainly concerned with server configuration options.

3. **Codeigniter :** Very easy starting setup, after copying the folders there is only base URL and database to be configured. Folder structure is ready to be filled with files by the user.

CODEIGNITER

’CodeIgniter is a powerful PHP framework with a very small footprint, built for PHP coders who need a simple and elegant toolkit to create full-featured web applications.
As a almost micro-framework is very simple to learn. Excellent documentation. Flexible and easy to use a party code.’

Advantages

1. Great for all size projects.
2. Super quick to learn and very simple to set up.
3. Excellent Documentation
4. Good collection of possessed libraries.
5. Easy to learn, adopt and deploy
6. Easy handling and customising.
7. Lightweight and extensive.
8. Configuration and normal coding rules are not required
9. Free from complex structures and development

Disadvantages

1. No Orm Support (Database mapping)
2. Its PHP based only and not very object-oriented in some parts
3. Company-driven instead of community-driven
4. Irregular releases

http://en.wikipedia.org/wiki/CodeIgniter

SYMFONY 2

’Symfony is a PHP web application framework for MVC applications. Symfony is free software and released under the MIT license. The symfony-project.com website launched on October 18, 2005
Uses the command line, yaml, powerful ORM. Proper view system, code generators, dependency injections for all. Very difficult to learn even though documentation is rather good to be fair.’

Advantages

1. Extremely power framework for big scale projects
2. Great documentation
3. High security framework
4. Faster and less greedy
5. Expandable

Disadvantages

1. Need to know OOP and console commands
2. Learn curve is very very high
3. Set up isn't beginner friendly

http://en.wikipedia.org/wiki/Symfony

PHP CAKE

‘CakePHP is an open source web application framework. It follows the Model-View-Controller (MVC) approach and is written in PHP, modeled after the concepts of Ruby on Rails, and distributed under the MIT License. CakePHP uses well-known software engineering concepts and software design patterns: convention over configuration, Model-View-Controller, Active Record, Association Data Mapping, and Front Controller. Lots of built-in functionality . Everything Is integrated rather closely. Documentations isn't in the best shape.’

Advantages

1. Install guide is super easy and quick to learn to use.
2. Extremely quick to learn
3. Great Orm Support (database mapping)
4. Helps reduce web application development cost & time considerably.
5. While classes can be challenging to work with in standard PHP, they are much easier to work with in Cakephp.
6. Its automated configuration process auto-detects preferred settings. What does this imply? This means one need not invest considerable time in configuring Linux-Apache-MySQL-PHP (LAMP) setup.

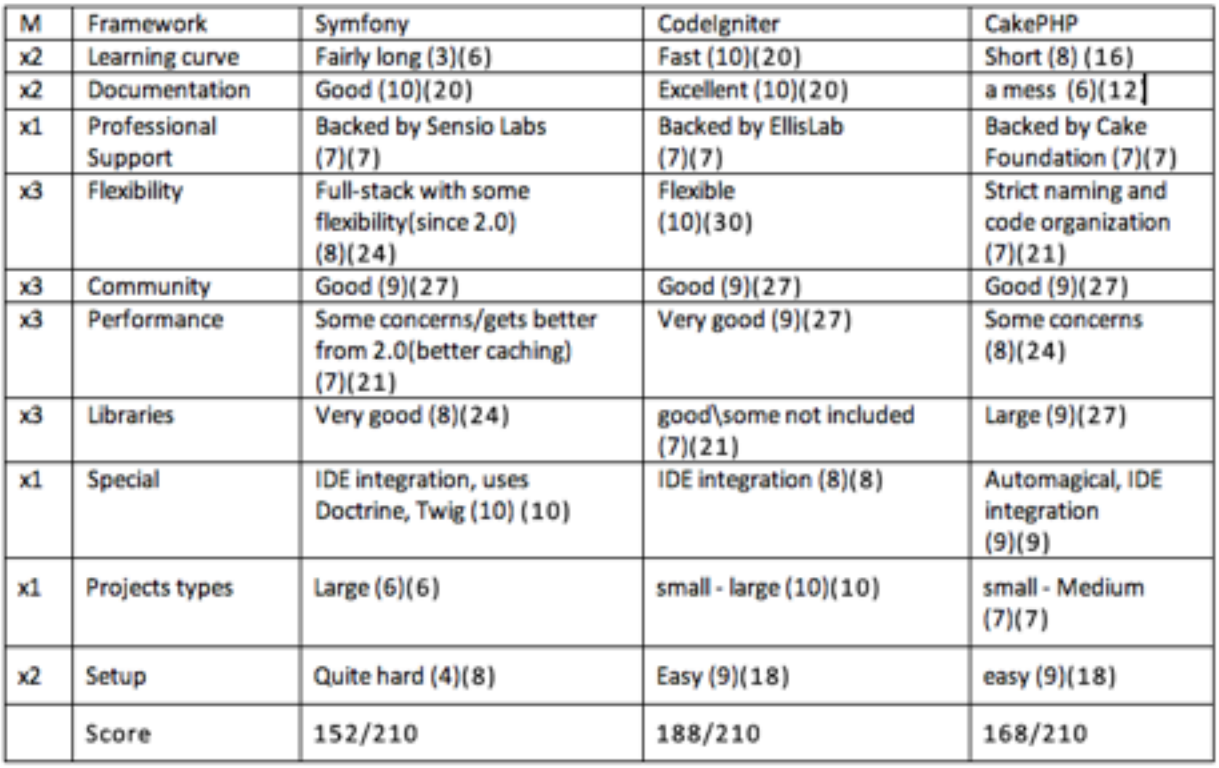
Disadvantages

1. Not great for large scale projects.
2. Documentations is a mess
3. While using CakePHP, one needs to update the default routes for creating Fancy URLs. If compared with other frameworks such as Symfony, CakePHP loses battle in this case.
4. Many still believe that CakePHP is easier to learn. But they haven’t come across frameworks such as CodeIgniter which make learning all the more easier!
5. One-way routing in CakePHP often proves to be a disadvantage when compared with frameworks such as Ruby on Rails.

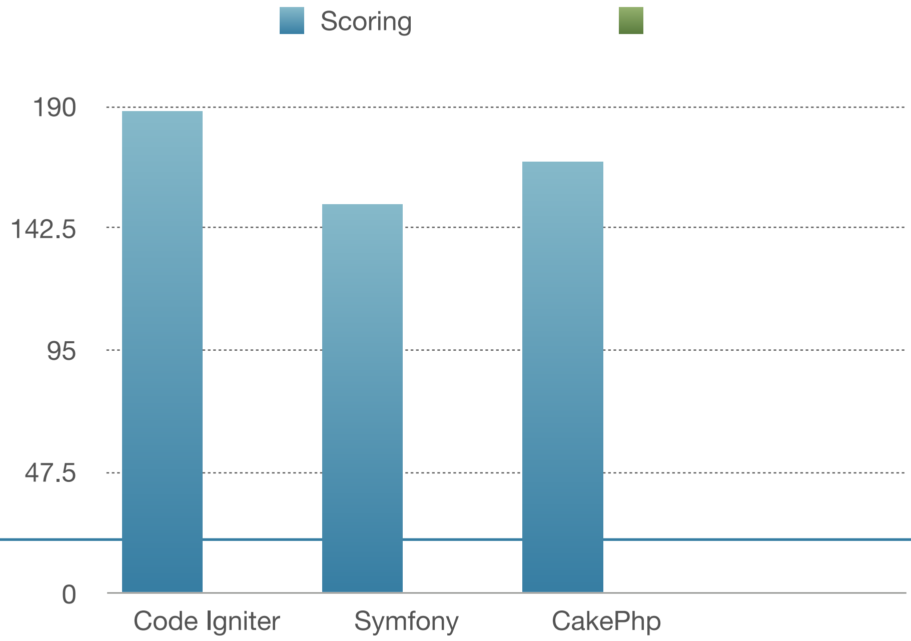
http://en.wikipedia.org/wiki/CakePHP

TABLES (SCORING)

For our scoring we are judging it on a scale of 0 - 10 . 0 being almost terrible and 10 being perfect wouldn't change a thing.



M	Framework	Symfony	CodeIgniter	CakePHP
x2	Learning curve	Fairly long (3)(6)	Fast (10)(20)	Short (8) (16)
x2	Documentation	Good (10)(20)	Excellent (10)(20)	a mess (6)(12
x1	Professional Support	Backed by Sensio Labs (7)(7)	Backed by EllisLab (7)(7)	Backed by Cake Foundation (7)(7)
x3	Flexibility	Full-stack with some flexibility(since 2.0) (8)(24)	Flexible (10)(30)	Strict naming and code organization (7)(21)
x3	Community	Good (9)(27)	Good (9)(27)	Good (9)(27)
x3	Performance	Some concerns/gets better from 2.0(better caching) (7)(21)	Very good (9)(27)	Some concerns (8)(24)
x3	Libraries	Very good (8)(24)	good\some not included (7)(21)	Large (9)(27)
x1	Special	IDE integration, uses Doctrine, Twig (10) (10)	IDE integration (8)(8)	Automagical, IDE integration (9)(9)
x1	Projects types	Large (6)(6)	small - large (10)(10)	small - Medium (7)(7)
x2	Setup	Quite hard (4)(8)	Easy (9)(18)	easy (9)(18)
	Score	152/210	188/210	168/210



CONCLUSION

A large dispute is in how models are created: CakePHP/Codeigniter models are written in PHP, and Symfony models are written in YAML.

CakePHP/Codeigniter ‘s approach is more similar to ROR's ActiveRecord (although it isn't exactly an AR implementation). CakePHP, in general, is more rails-esque.

CakePHP's and Codeigniter’s documentation and tools, in my opinion, have a wider target audience and the syntax and helpers are easier, but they have yet to embrace PHP5 as their exclusive target (to autoloading isn't really there). In general, I prefer codeigniter's approach because it sort of follows an established standard, and I applaud its organization.

CakePHP also offers dead simple to understand tools. Symphony's'CRUD' in just isn't the same. Combine that with Symphony's (awkward) action syntax and throw in Symphony's poorly designed (and challenging to understand) documentation, and preference for 3rd-party paid documentation (books on Amazon) and you have more ticks in the cons column.

So which framework is better ? well it depends on our choice for the project and the scale of it . Just even Googling which is better will have different results depending on people's preferences.

So that being said, we picked code ighter for the group project because of the following reasons we took into account . That the project is small in scale, we want a framework that is easy to learn and has features and benefits that will make this easier for us.

PROJECT PROPOSAL

The app is to pick up the users location load. The user can then put in their destination , it will search google maps and bring them to another page . Where it will display the weather in that location along with photos from flickr of that destination and a map of all the ways the user can get into the destination, users can save the travel event to google calendar.

Requirements

1. Build profile of user to tailor the search results to their interests.
2. Picks up/requests your location
3. Tells you all the ways to get there (bus,map, trains etc)
4. The weather for your arrival point
5. Flickr results for the destination

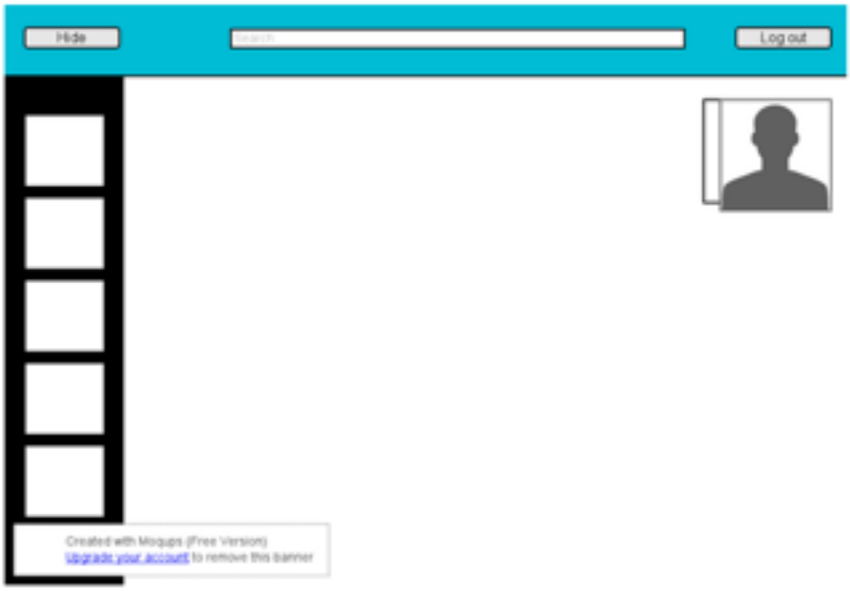
Who will use the website.

1. Anyone from the age of 15 - 99 who is traveling.

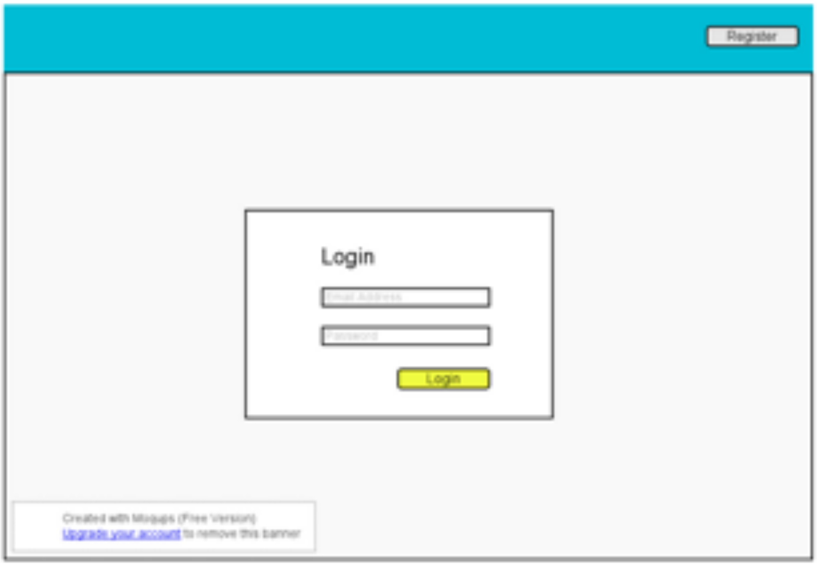
Frameworks and apis used

1. Flickr Search api
2. Yahoo weather app
3. Google maps api
4. Google calendar.
5. Codeigniter for the framework

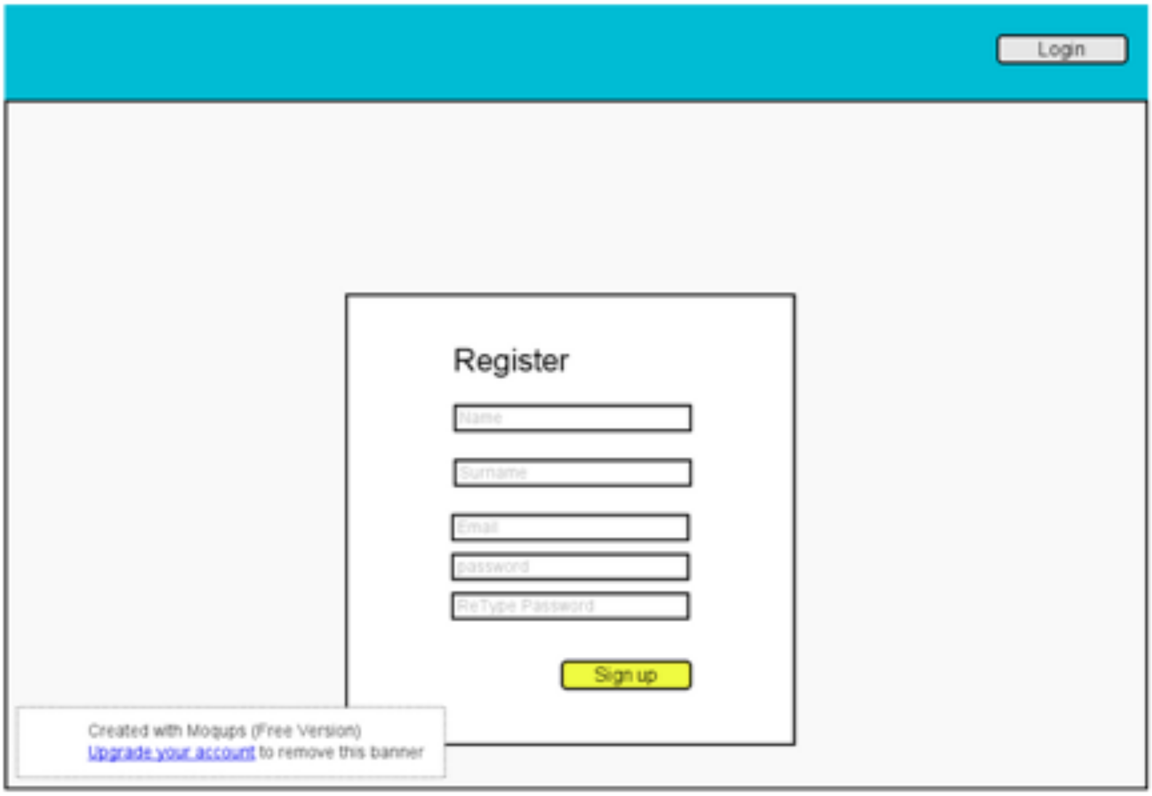
SCREENSHOT



Created with Moqups (Free Version)
Upgrade your account to remove this banner



Created with Moqups (Free Version)
Upgrade your account to remove this banner



Created with Moqups (Free Version)
Upgrade your account to remove this banner