



Multi Language Content

Software Requirements Specification

for

Google Sommer of Code 2008

Updated on: 20.03.2008

Inhaltsverzeichnis

1. Introduction.....	3
1.1. About this document.....	3
1.2. Technicalities.....	3
2. Definition of objectives.....	4
2.1. Actual state and motivation.....	4
2.2. Must have criteria.....	4
2.3. Should have criteria.....	4
2.4. Functional constraints.....	4
3. Design goals.....	5
3.1. Maintainability.....	5
3.2. Reusability.....	5
3.3. Usability.....	5
4. Development aspects.....	6
4.1. Used tools.....	6
4.2. Communication with the mentor.....	6

1 Introduction

1.1 About this document

This is a requirements specification for the project „Multi Language Content“ within the participation of the PostNuke Application Framework in Google Summer of Code 2008.

This document does not contain specific details like use cases, service conditions or implementation aspects. It lists up the most important aspects, but it can be mostly seen as a rough guideline based on our own ideas. These are not obligatory, we are also looking forward to getting innovative input from the students.

1.2 Technicalities

- Project: Multi Language Content
- Difficulty: easy
- Required experience: PHP, MySQL, Language structure

This project will be mentored by {MENTOR NAME}. Please find more information about communication between mentors and students at chapter 4.2 on page 6.

2 Definition of objectives

2.1 Actual state and motivation

Every system used in professional environments needs a way has to support multilingualism. PostNuke does that in general and the language system is even being improved another time to support gettext in future.

Although PostNuke supports multi-language content in some core modules, there is no API provided by the framework for third party developers to implement mutli language content in their work. This results in different ways to handle this in a module-specific way.

2.2 Must have criteria

The aim of this project is implementing that API and provide flexible means to make data objects supporting multiple languages. This could happen by associating all translations of a particular content item together and providing a consistent translations interface.

Particular attention would need to be paid to the front end implementation of this system, to ensure ease of use and simple integration with other modules.

Anyone taking on this project would need to coordinate with the developer of the project „Revisions Control“.

2.3 Should have criteria

Additionally a queue of items and translation task assignment for large websites would be useful for sites with a translations team.

2.4 Functional constraints

The project only aims on dynamical content objects. All other areas are covered by the normal language system.

3 Design goals

3.1 Maintainability

The whole source code should follow the PostNuke coding standards and contain some comments for improved understandability.

Ideally some pattern are used during the design phase for reducing the amount of physical maintenance points.

3.2 Reusability

The system should work independent on particular assumptions about the treated data objects resulting in a greater reusability for several different modules. For example an interface type could be used in operative code ensuring exchangeability through polymorphism. This way module developers would only have to implement a subclass definition their module-specific data structures.

3.3 Usability

As pointed out the frontend design is important for integration into possible use cases. An easy and intuitive user guidance is necessary for acceptance on the part of the community. If possible within the given time frame a small documentation could be written describing functionality and usage.

4 Development aspects

4.1 Used tools

There are no particular requirements. In general the following software components are needed for this project:

- Webserver (e.g. Apache)
- Database (e.g. MySQL, PostgreSQL, MS SQL or Oracle)
- PHP 5
- Latest version of PostNuke .8
- Programming editor (either a comprehensive IDE like Eclipse with PDT or a text editor with syntax highlighting)
- Dictionaries :-)

4.2 Communication with the mentor

We created a mailing list where general things can be discussed between all mentors and students. The mentors are looking forward to some exciting projects.

If you have any further questions please contact the PostNuke Steering Committee at sc@postnuke.com.