Universita' degli Studi di Messina
Dipartimento di Matematica e Informatica

Database course project

veeForum

29 May 2015

Author:
Vittorio Romeo

Professor:
Massimo Villari

http://vittorioromeo.info

http://unime.it

Contents

Part I

Project specifications

The following part of the document describes the project and its design/development process without exploring its implementation details.

The part begins with a synthesis of the **client request**, which is then analyzed into a **Software Requirements Specification** (SRS).

Writing a correct and informative SRS is utmost importance to achieve an high-quality final product.

The SRS will cover the following points in depth:

- **Scope and purpose**.
- **Feature and functions**.
- **External interface requirements**.
- **Functional requirements**.
- **Example use cases**.
- **Non-functional requirements**.
- **Analysis models**.

Chapter 1

Client request

The client requests the design and implementation of a **forum creation/management framework** and a **modern responsive web forum browsing/management application**.

The client intends using the requested forum framework **to build communication platforms** for various projects, both for internal employee usage and interaction with the public.

It is imperative for the system to allow administrators to easily well-organized create **content-section hierarchies** and **user-group hierarchies**.

Administrators also need to be able to **give groups specific permissions for every section**.

Some sections will only be visible and editable to employee groups (e.g. internal discussion), some sections will be visible but not editable by the public (e.g. announcements), and others will need to be completely open to the public (e.g. technical support).

Being able to **keep track of user-created content** is also very important for the client.

Initially, tracking the date and the author of the content will be enough, but the system has to be designed in such a way that inserting additional creation information (e.g. browser/operating system used to post) will be trivial.

In the future, additional content types (e.g. videos, attachments) may be added to the system and their creation will have to be tracked as well.

This data needs to be independent from the contents, in order to easily allow administrators and project managers to gather statistical data on forum usage.

The web application has to be extremely simple but flexible as well. Administrators need be able to perform all functions described above through a responsive admin panel.

Content consumers and creators should be able to view and create content from the same responsive interface.

Moderators and administrators should be able to edit and delete posts through the same interface as well. User interface controls will be shown/hidden depending on the users

permissions.

Chapter 2

Software Requirements Specification

1. Introduction

This **Software Requirements Specification** (SRS) chapter contains all the information needed by software engineers and project managers to design and implement the requested forum creation/management framework.

The SRS was written following the **Institute of Electrical and Electronics Engineers** (IEEE) guidelines on SRS creation.

1..1 Purpose

The SRS chapter is contained in the **non-technical** part of the thesis.

Its purpose is providing a **comprehensive description** of the objective and environment for the software under development.

The SRS fully describes **what the software will do** and **how it will be expected to perform**.

1..2 Scope

1..2.1 Identity

The software that will be designed and produced will be called **veeForum**.

1..2.2 Feature extents

The complete product will:

- Provide a framework for the **creation and the management of a forum system**.

- Allow its users to **deploy and administrate** multi-purpose forums.
- Give access to a **modern responsive web application** to setup, browse and manage the forum.

veeForum, however, will not:

- Provide infrastructure or implementation for a complete blog/website. The scope of the software is forum building.
- Implement instant private messaging - user-to-user chat is beyond the scope of the project.

1..2.3 Benefits and objectives

Deploying veeForum will give its users a number of important benefits and will fulfill specific objectives.

- Companies and individuals making use of veeForum will have access to an **easy-to-install** and **easy-to-use** forum creation and management platform.
- Users and moderators of the deployed forums will be able to **easily create, track and manage** content and other forum users.
- Forum administrators will be given **total control** of the forum structure, users and permissions through an **easy-to-use** responsive administration panel.

1..3 Definitions, acronyms and abbreviations

aaa

1..4 Overview

aaa

2. General description

aaa

2..1 Product perspective and functions

The product shares many basic aspects and features with existing forum frameworks such as **phpBB** or **vBulletin**: flat/threaded discussion support, nested sections, user attachments, etc.

veeForum improves on existing forum frameworks in the following ways:

- Provides a responsive web interface without postbacks.
- Allows users and moderators to subscribe and unsubscribe not only to posts, but to users and sections as well.
- Has a powerful real-time Facebook-like notification system that notifies users when tracked content has been added or edited.
- Gives administrator the possibility to design and manage complex permission hierarchies for user groups and single users.

2..2 User characteristics

veeForum needs to target both users that **only consume the content offered by deployed forums**, users that **actively create and manage content in deployed forums**, and users that **build and deploy forum instances**.

User-friendliness is essential for every target, but all the required functionality is effectively exposed to different user groups.

It is therefore required to have clear interfaces thar do not negatively affect the user experience by being either too complex or too simple (all features need to be exposed).

3. Specific requirements

aaa

3..1 External interface requirements

aaa

3..1.1 User interfaces

The product will provide both a desktop and a mobile user web interface.

- **Web interface**: it is required to provide a modern responsive web interface, compatible and tested with the most popular browsers (Internet Explorer 10+, Google Chrome, Mozilla Firefox). The web interface will give forum access to users and moderators, and administrator access to forum management staff.

- **Mobile interface**: is is required to provide a modern mobile application for the major platforms (Android, iOS, Windows Phone). The mobile application will allow browsing and content management of forums created with the product.

3..1.2 Software interfaces

The **open-source policy** of veeForum will allow framework users to expand or improve existing functionality and to interact with other existing technologies.

Accessing and modifying forum data will be possible through **RESTful** requests, returning and accepting **JSON** (Javascript Object Notation).

3..2 Functional requirements

In software engineering, a **functional requirement** defines a function of a system and its components.

Functional requirements may be **calculations**, **technical details**, **data manipulation and processing** and other specific functionality that define what a system is supposed to accomplish.

Behavioral requirements describing all the cases where the system uses the functional requirements are captured in **use cases**.

3..2.1 User/group management

- **Users**: users will be managed by the system. Users can register (or be manually added by an administrator). Registration can be configured to require a confirmation email or not.

- **Groups**: every user will be part of at least one group at all times. Groups are part of an hierarchy: they can inherit from each other. Groups can have permissions specific to sections and system-wide permissions.

3..2.2 Content hierarchy

- **Posts**: posts will be the base of the content hierarchy. They will contain HTML-enabled text and any number of attachments. Posts can be edited and deleted by the original owner.

- **Threads**: threads are groups of posts. Users with the correct permissions can create a thread in a specific section and have other users add posts or subscribe to it. Threads can be edited and deleted by the original owner.

- **Sections**: sections are content containers intended to group threads related to the same subject. Forum administrators and moderators can create sections and give users permissions to view or edit them.

3..2.3 Content tracking system

- **Creation data**: user-created content (posts, threads,attachments, etc) will have some data specific to its creation can be extended by forum administrators. Basic predefined data will consist of creation date and time. It will be possible to run statistical queries on content creation data.

- **Subscriptions**: users and moderators will be able to subscribe to specific sections, threads or user to track their contents. They will receive real-time notifications upon addition/editing of tracked content.

3..3 Example use cases

In software and systems engineering, a **use case** is a list of steps, typically defining interactions between one or more actors and a system, to achieve a goal.

3..3.1 Mobile game forum

A company developed a popular mobile game, with a wide audience. The company uses the veeForum framework to give users a place to discuss game strategy, give feedback on the quality of their product and receive technical support.

3..3.1.1 Actors

- Game developers.

- Game players.

- Forum management team.

- Technical support team.

- Feedback (PR) team.

3..3.1.2 Pre-conditions

- Release of a popular product with a wide audience.
- Game users need to register on the forum.

3..3.1.3 Flow of events

- Installation and configuration of a veeForum-enabled forum system by the forum management team.
- Creation of the sections and permission hierarchies by the forum management team.
- Registration and content creation by the game developers and game players.

3..3.1.4 Post-conditions

- Game players will be able to share their strategies and thoughts on the product.
- The technical support team will find all technical issues grouped in a convenient way and will be able to track individual issues. Technical support members will be able to communicate with each other in a private section.
- The feedback team will be able to track user suggestions and forward potential product improvements to the developer team.

3..4 Non-functional requirements

Functional requirements are supported by **non-functional requirements** (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

3..4.1 Performance

The system will be designed from the ground-up with emphasis on performance. As the forum may have huge amounts of contents and concurrent usage after its deployment, optimizing is a must.

When possible, functions will be implemented **directly in the database**, for maximum performance.

Web backend functions will also be carefully **optimized both for memory and speed**.

3..4.2 Reliability

The system will have to be reliable and keep working in case of errors.

Database queries and functions will be executed in **safe wrappers** that catch and handle errors carefully.

3..4.3 Security

veeForum needs to guarantee privacy and security for users and administrator of the system.

Well-tested and well-received **security idioms** and **encryption algorithms** will have to be used throughout the implementation of the whole system.

3..4.4 Maintainability and portability

Being an open-source project, **maintainability**, **extensibility** and **portability** are key.

The code layer will be carefully designed and organized to allow easy maintenance, bug-fixing and feature addition.

To ensure maximum portability, the product will be designed to work on the most popular **GNU/Linux** distributions and will be thoroughly tested on different platforms.

3..5 Logical Database Requirements

aaa

4. Analysis models

aaa

4..1 Sequence Diagrams

aaa

4..2 Data Flow Diagrams

aaa

4..3 State-Transition Diagrams

aaa

Part II

Technical analysis

The following part of the thesis will cover all implementation choices and details for veeForum in depth.

Firstly, the **development environment and tools** and **chosen technologies** will be described and motivated.

Afterwards, the technical details, including code examples and APIs, will be described for the two modules of the application: the **database** and the **web application**.

Every **table** of the database will be analyzed in detail, directly showing commented **DDL** code. The database also contains important **stored procedures** and **triggers** that are core part of the system's logic and that need to be explained in depth - the related **DML** code will be shown and commented.

The web application itself is divided in multiple modules:

- A **database interface backend module**, that interfaces with the database and wraps its tables and stored procedures.
- A **HTML5 generation module**, that greatly simplifies the creation of dynamic forum web pages by wrapping HTML5 controls in **object-oriented wrappers** that can be easily bound to callbacks and database events.
- A **modern responsive AJAX frontend** that allows users and interact with the backend module from multiple device, limiting postbacks and page refreshes.

Chapter 3

Development process

1. Environment and tools

All modules of veeForum have been developed on **Arch Linux x64**, a lightweight GNU/Linux distribution.

Arch is installed as a minimal base system, configured by the user upon which their own ideal environment is assembled by installing only what is required or desired for their unique purposes. GUI configuration utilities are not officially provided, and most system configuration is performed from the shell and a text editor. Based on a rolling-release model, Arch strives to stay bleeding edge, and typically offers the latest stable versions of most software.

No particular integrated development environments (IDEs) were used during the development - a modern graphical text editor, **Sublime Text 3**, was used instead.

2. Docker

Docker is an open-source project that **automates the deployment of applications** inside software containers, by providing an additional layer of abstraction and automation of operating-system-level virtualization on Linux.

Docker uses resource isolation features of the Linux kernel such as **cgroups** and **kernel namespaces** to allow independent containers to run within a single Linux instance.

This technology has been used since the beginning of the development process to **separate veeForum data and packages** from the host system and to dramatically increase **portability** and **ease of testing**.

Docker is also used for the installation of the product on target systems - with a single command it is possible to **retrieve all required dependencies**, correctly **configure the system** and **automatically install veeForum**.

3. Version control system

Version control systems (VCSs) allow the **management of changes** to documents, computer programs, large web sites, and other collections of information.

Nowadays, a version control system is **essential** for the development of any project. Being able to track changes, develop features in separate **branches**, have multiple programmers work on the same code base without conflicts and much more is extremely important for projects of any scope and size.

The chosen VCS is **Git**, a distributed revision control system with an emphasis on **speed**, **data integrity**, and support for **distributed, non-linear workflows**.

Git is widely appreciated in the private and open-source programming communities - it was initially designed and developed by **Linus Torvalds** for Linux kernel development in 2005, and has since become the most widely adopted version control system for software development.

The veeForum project is **open-source** and **appreciates feedback and contributions**. It is hosted on **GitHub**, a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git, while adding **additional features** that make collaboration and public contributions easy and accessible.

4. Apache

The Apache HTTP server is the world's most widely used web server software.

Apache has been under open-source development for about 20 years - it supports all modern server-side technologies and programming languages, and also is **extremely reliable** and **secure**.

5. Thesis

The current document was written using LaTeX, an high-quality typesetting system; it includes features designed for the production of **technical and scientific documentation**.

LaTeXwas chosen for the current document because of the visually pleasant typography, its extensibility features and its abilities to include and highlight source code.

5..1 LatexPP

A small **C++14** LaTeXpreprocessor named **LatexPP** was developed for the composition of this thesis.

LatexPP allows to use an intuitive syntax that avoids markup repetition for code highlighting and macros.

Preprocessing and compiling a LaTeXdocument using LatexPP is simple and can be automated using a simple **bash** script.

```
1  #!/bin/bash
2
3  latexpp ./thesis.lpp > ./thesis.tex
4  pdflatex -shell-escape ./thesis.tex && chromium ./thesis.pdf
```

Chapter 4

Project structure

aaa

1. Folder structure

aaa

2. PHP Module

aaa

3. SQL Module

aaa

4. Other data

aaa

Chapter 5

Installation

aaa

Chapter 6

Conceptual model

aaa

Chapter 7

Logical model

aaa

Chapter 8

Table details

aaa

Chapter 9

Web interface

aaa

Chapter 10

Sample queries

aaa

Part III

Conclusion

aaa

Chapter 11

Final product

aaa

Chapter 12

What I learned

aaa

Chapter 13

Future

aaa

Chapter 14

References

aaa