Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 28/Nov/2013 | 1.0 | Initial software architecture description | Storm Cloud Development |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Architectural Representation 4

3. Architectural Goals and Constraints 4

4. Use-Case View 5

4.1 Use-Case Realizations 5

5. Logical View 5

5.1 Overview 5

5.2 Architecturally Significant Design Packages 5

6. Process View 6

7. Deployment View 6

8. Data View (optional) 6

9. Size and Performance 7

10. Quality 7

# Introduction

## Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

## Scope

This Software Architecture Document provides an architectural overview of the ProjectCM web application. This Document has been partly generated directly out of Ruby on Rails.

## Definitions, Acronyms, and Abbreviations

MVC Model-View-Controller

SRS Software Requirements Specification

SAD Software Architecture Document

## References

SRS Software Requirements Specification.pdf

## Overview

The SAD shows how the architecture of the software is. So we will start with the general representation and then show the goals and constraints. Afterwards the use cases and the different views are explained. These views are the logical view, the process view, the deployment view and the data view. In the end there is a paragraph about size and quality.

# Architectural Representation

The software architecture is organized as MVC. So this document contains information about model, view and controller.

# Architectural Goals and Constraints

## Security

All important data e.g. the password has to be stored as a hash. So it should not be possible to use this data when someone who should not have access to it gains access. Also all authorization checks must be done by the server to avoid manipulations by the client. It should be possible fetch emails with SSL from other servers to the ProjectCM server and in production environments all communication between client and server must be via SSL.

## Distribution / Reuse / Dependencies

The application is on a single instance when we start with it, but it should be possible to scale it on different servers. Also as much code as possible should be reused. There are no external dependencies.

# Use-Case View

See SRS

## Use-Case Realizations

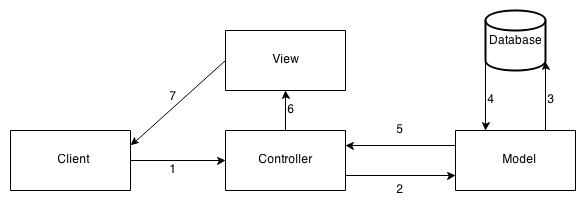
See SRS

# Logical View

A description of the logical view of the architecture. Class diagrams may be included to illustrate the relationships between architecturally significant classes, subsystems, packages and layers.

[This section describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages. And for each significant package, its decomposition into classes and class utilities. You should introduce architecturally significant classes and describe their responsibilities, as well as a few very important relationships, operations, and attributes.]

## Overview



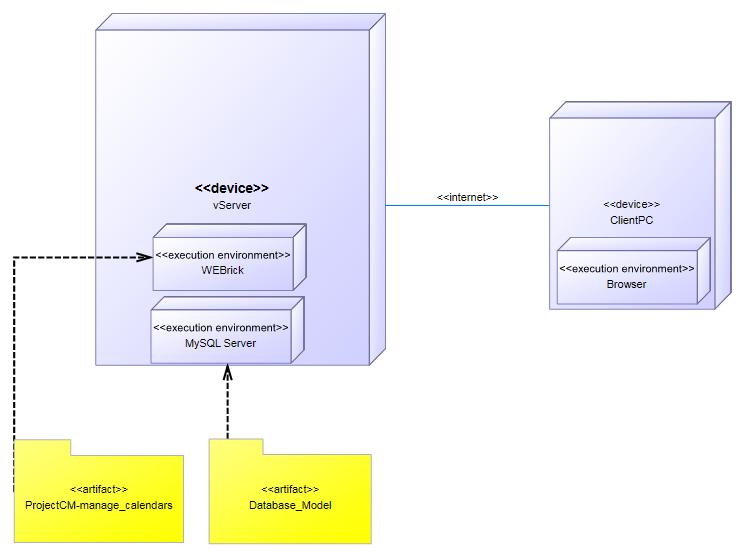
The client sends a request to the Controller (1). The controller calls the model to prepare the data (2). The model fetches the data from the database (3, 4), prepares it and returns the prepared data to the controller (5). The controller chooses the view to show the data with and sends the data to the view (6). The view shows the page to the client (7).

## Architecturally Significant Design Packages

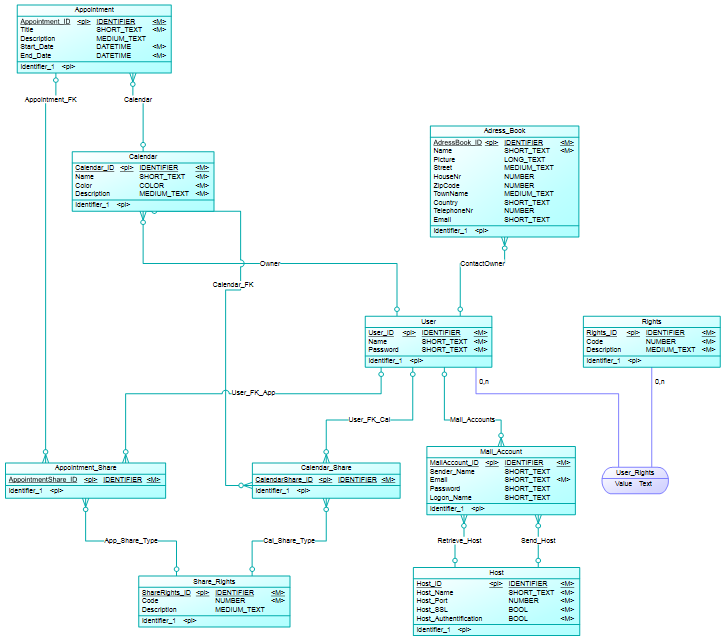
# Process View

n/a

# Deployment View



# Data View



# Size and Performance

n/a

# Quality

n/a