ProjectCM

Software Architecture Document

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

Software Architecture Document

# Introduction

[The introduction of the **Software Architecture Document** provides an overview of the entire **Software Architecture Document**. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the **Software Architecture Document**.]

<http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm>

## 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

SRS Software Requirements Specification

## References

SRS Software Requirements Specification.pdf

## Overview

[This subsection describes what the rest of the **Software Architecture Document** contains and explains how the **Software Architecture Document** is organized.]

# Architectural Representation

[This section describes what software architecture is for the current system, and how it is represented. Of the **Use-Case**, **Logical**, **Process**, **Deployment**, and **Implementation Views**, it enumerates the views that are necessary, and for each view, explains what types of model elements it contains.]

# Architectural Goals and Constraints

[This section describes the software requirements and objectives that have some significant impact on the architecture; for example, safety, security, privacy, use of an off-the-shelf product, portability, distribution, and reuse. It also captures the special constraints that may apply: design and implementation strategy, development tools, team structure, schedule, legacy code, and so on.]

Security: passwort verschlüsselt, berechtigungen serverseitig prüfen, möglichst viel reuse durch ruby, per ssl mails abrufen serverseitig, ggf per ssl zugriff auf anwendung, single instance, skalierbarkeit

No external dependencies.

Write here which external interfaces and dependencies you have.

Make sure to include external systems and libraries in your diagrams below.

# Use-Case View

See SRS

## Use-Case Realizations

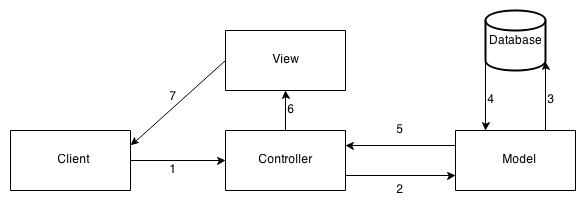
See SRS

# 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

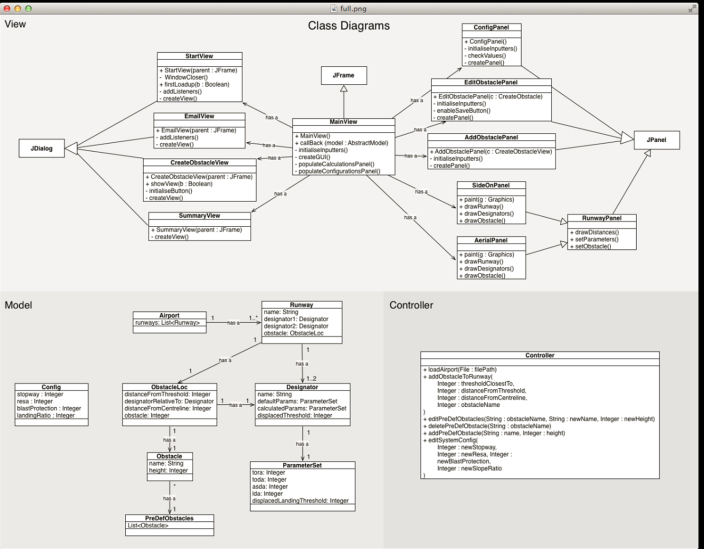


The client sends a request to the Controller (1). The controller calls the model to prepare the data (2). The model fetches the data from the database (3, 4), prepares it and returns the prepared data to the controller (5). The controller chooses the view to show the data with and sends the data to the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

## Architecturally Significant Design Packages

[For each significant package, include a subsection with its name, its brief description, and a diagram with all significant classes and packages contained within the package.

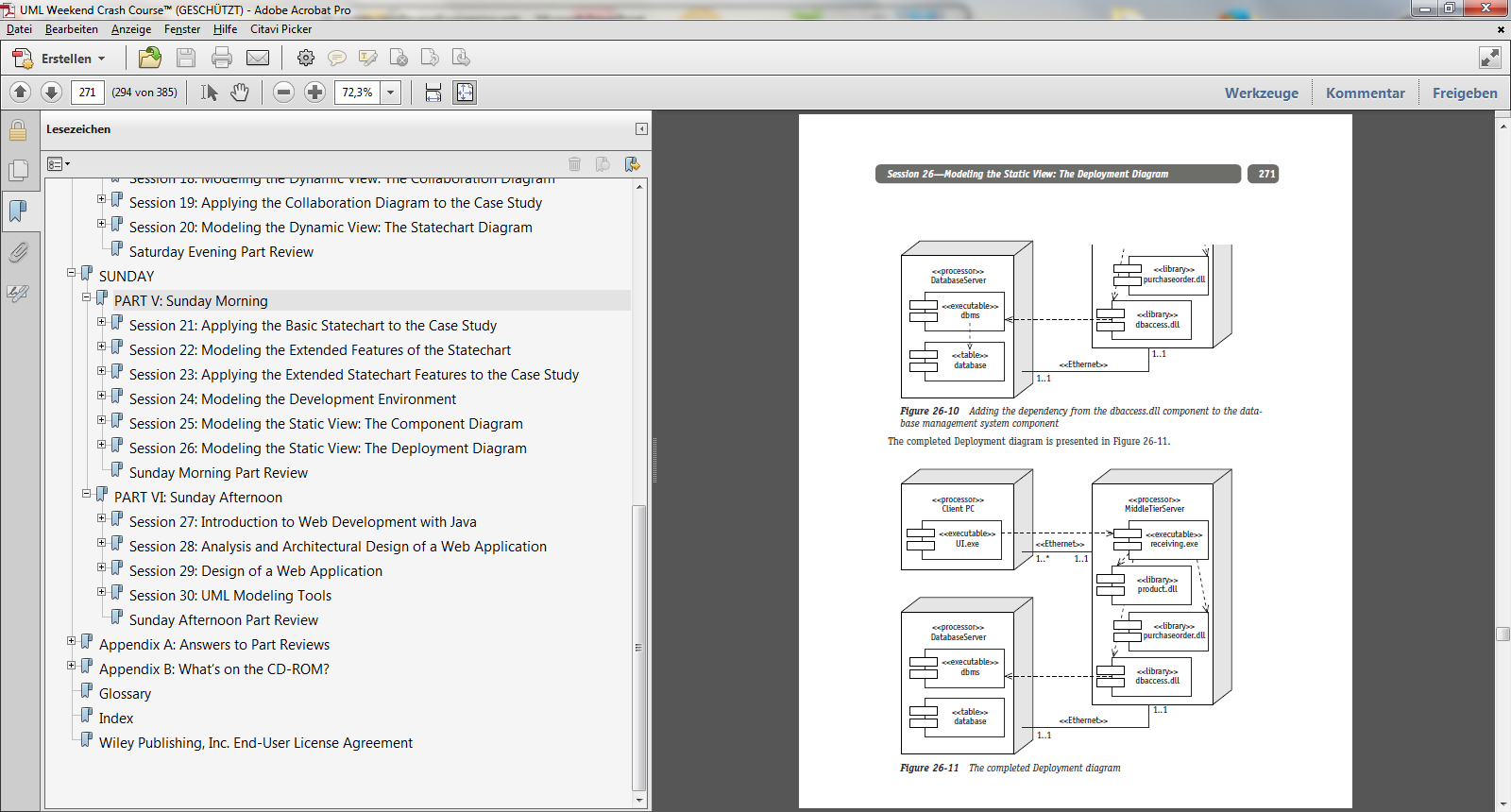
For each significant class in the package, include its name, brief description, and, optionally, a description of some of its major responsibilities, operations, and attributes.]



# Process View

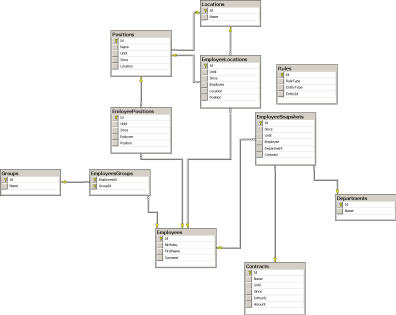
n/a

# Deployment View

**

# Data View (optional)

[A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data, or the translation between the Design Model and the Data Model is trivial.]



# Size and Performance

n/a

# Quality

n/a