SocialFunnel

Version 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 14/11/2014 | 1.0 | First Version | Daniel Betsche |
| 31/05/2015 | 1.1 | Pattern Added | Daniel Betsche |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 1

2. Architectural Representation 1

3. Architectural Goals and Constraints 1

4. Use-Case View 1

4.1 Use-Case Realizations 1

5. Logical View 1

5.1 Overview 1

5.2 Architecturally Significant Design Packages 1

6. Process View 1

7. Deployment View 1

8. Implementation View 1

8.1 Overview 1

8.2 Layers 1

9. Data View (optional) 1

10. Size and Performance 1

11. Quality 1

# Introduction

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.

Das Dokument gibt einen verständlichen Überblick über die Architektur des Systems. Verschiedene

Sichtweisen der Architektur dienen dazu um unterschiedliche Aspekte des Systems abzubilden.

Es ist dazu gedacht, die signifikanten Entscheidungen, die zur Architektur des Systems getroffen wurden, zu erfassen und zu übermitteln.

# Architectural Representation

Die Software wird nach dem MVC-Patter (Model View Controller) aufgebaut um eine flexible Umgebung zu schaffen. Das Aufteilen der Software in 3 eigenständige Zuständigkeitsbereiche erlaubt ein sauberes und schnelles Austauschen der Module. Dabei wird unterschieden zwischen der Geschäftslogik, dem Controller, dem Backend welches die Datenbank beinhaltet, dem Model und der Benutzeroberfläche, dem View.

# Architectural Goals and Constraints

Das System verwendet Spring –MVC zur Umsetzung des Patterns.

Spring bietet dabei eine Vielzahl an Erweiterungen und Bibliotheken um das Abrufen, Darstellen und Verarbeiten von Daten zu erleichtern. Folgende Bibliotheken werden in diesem Projekt benutzt

* Controller: Spring, Spring Security
* Model: Hibernate
* View: Spring Web Flow, Vaadin

# Use-Case View

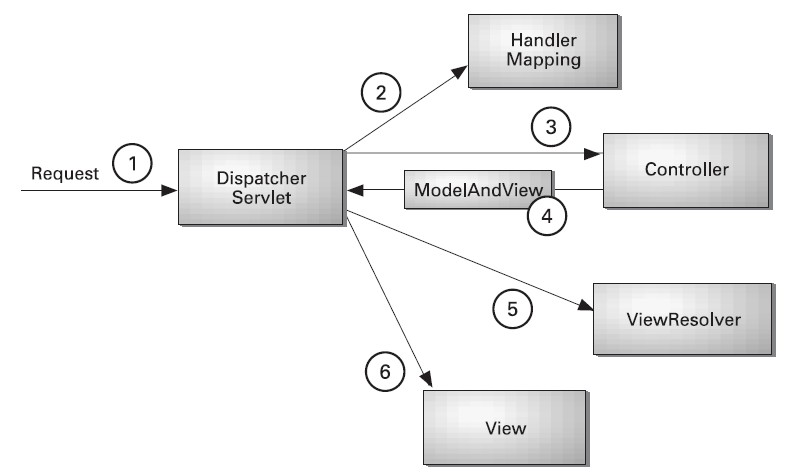
(n/a)

## Use-Case Realizations

(n/a)

# Logical View

## Overview

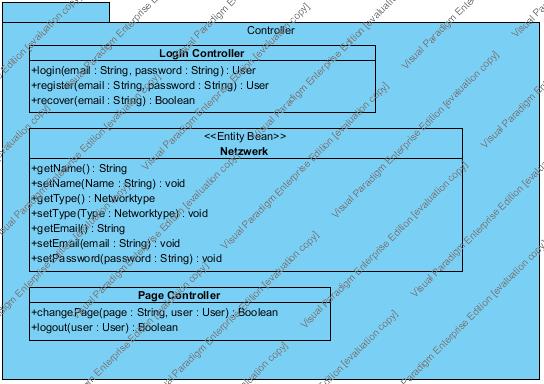


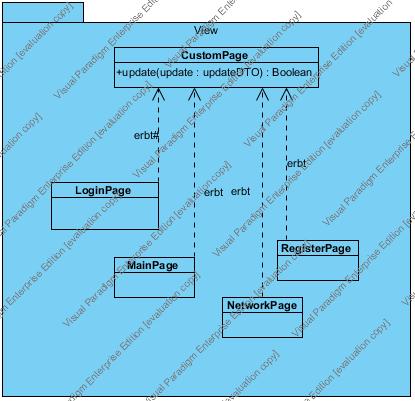
In combination with

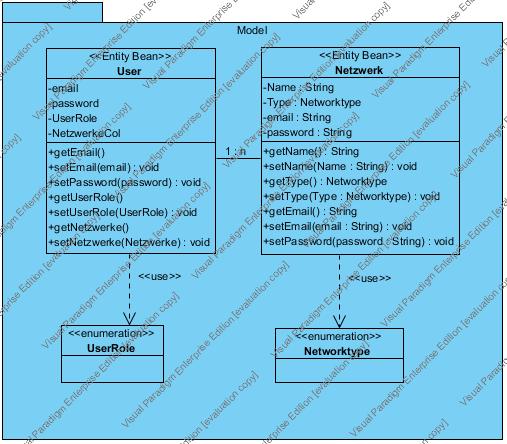


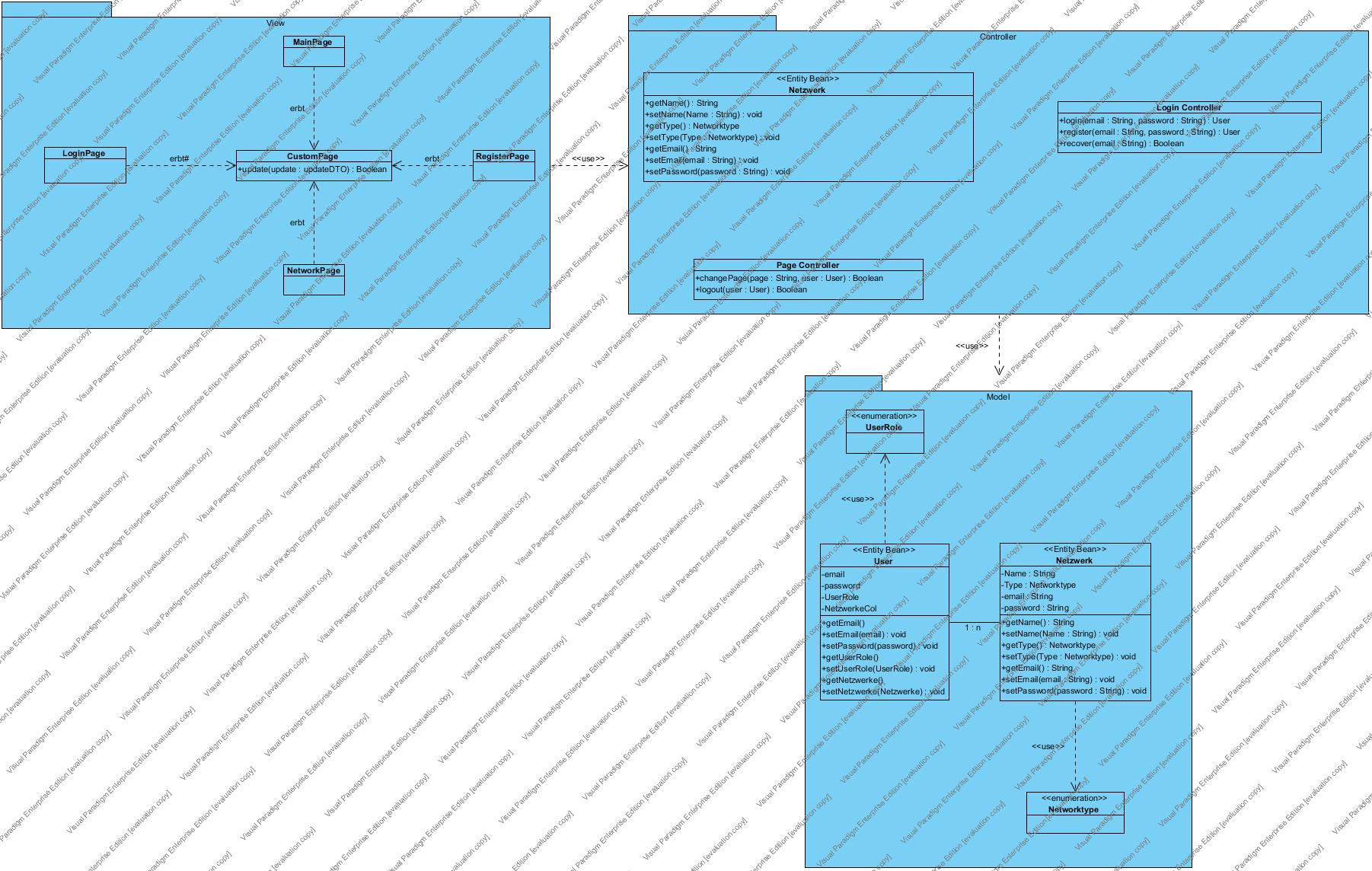
Quelle: Vaadin.com

## Architecturally Significant Design Packages







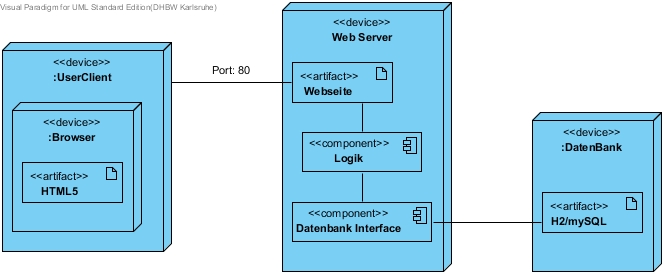


Proxy Pattern implemented: [link to Pattern document](https://docs.google.com/document/d/1dcQ-KOfKMRPD-eKg7mHaTCtIT9bRa0UmknGeVcz2k00)

# Process View

(n/a)

# Deployment View



# Implementation View

(n/a)

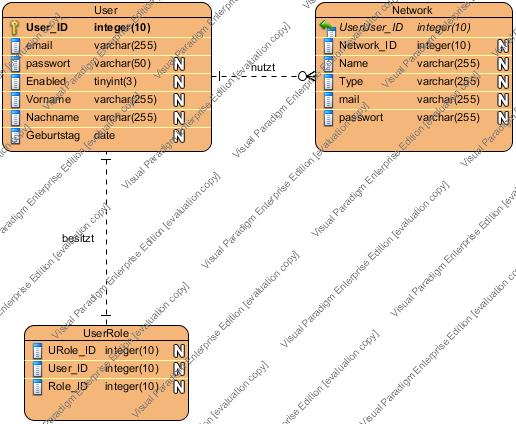
## Overview

(n/a)

## Layers

(n/a)

# Data View (optional)



# Size and Performance

(n/a)

# Quality

(n/a)