

LocalSharing Software Architecture Document

Version <1.0>

LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>

Revision History

Date	Version	Description	Author
11/Nov/2014	1.0	Initiale Version	Julia Anken Johannes Grün Corinna Hertweck
13/Nov/2014	1.0	Textuelle Beschreibung	Julia Anken
13/Nov/2014	1.0	Logical View/ Deployment View Diagrams	Johannes Grün
13/Nov/2014	1.0	DB Model	Corinna Hertweck
17/Nov/2014	1.0	Logical View - Overview	Julia Anken
17/May/2015	1.0	Aktualisierung der Diagramme	Julia Anken
01/Jul/2015	1.0	Aktualisierung der Diagramme für final Handin	Julia Anken

LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>

Table of Contents

1.	Introduction	4
2.	Architectural Representation	4
3.	Architectural Goals and Constraints	4
4.	Use-Case View	4
4.1	Use-Case Realizations	4
5.	Logical View	5
5.1	Overview	5
5.2	Architecturally Significant Design Packages	6
6.	Process View	12
7.	Deployment View	12
8.	Implementation View	12
8.1	Overview	12
8.2	Layers	12
9.	Data View (optional)	13
10.	Size and Performance	13
11.	Quality	13

LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>

Software Architecture Document

1. Introduction

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.

2. Architectural Representation

Zur Strukturierung wird MCV (Model-View-Controller) verwendet. Dies sagt aus, dass es die 3 Einheiten Model, View und Controller gibt. Das Ziel dahinter ist ein flexibler Programmentwurf, sodass spätere Änderungen und Erweiterungen leicht möglich sind und einzelne Komponenten wiederverwendet werden können.

3. Architectural Goals and Constraints

In dem System wird Spring MVC verwendet.

Das Spring Framework bietet ein umfassendes Programmier- & Konfigurationsmodell für Java-basierte Enterprise Applications auf jeder Art von Deployment-Plattform. Ein Feature des Frameworks sind Spring MVC Webapplications, die das MVC-Pattern implementieren.

Folgendes wurde für das MCV-Pattern im System verwendet:

- Controller: Spring
- Model /DataAccess: Hibernate
- View: Thymeleaf Template Engine

4. Use-Case View

(n/a)

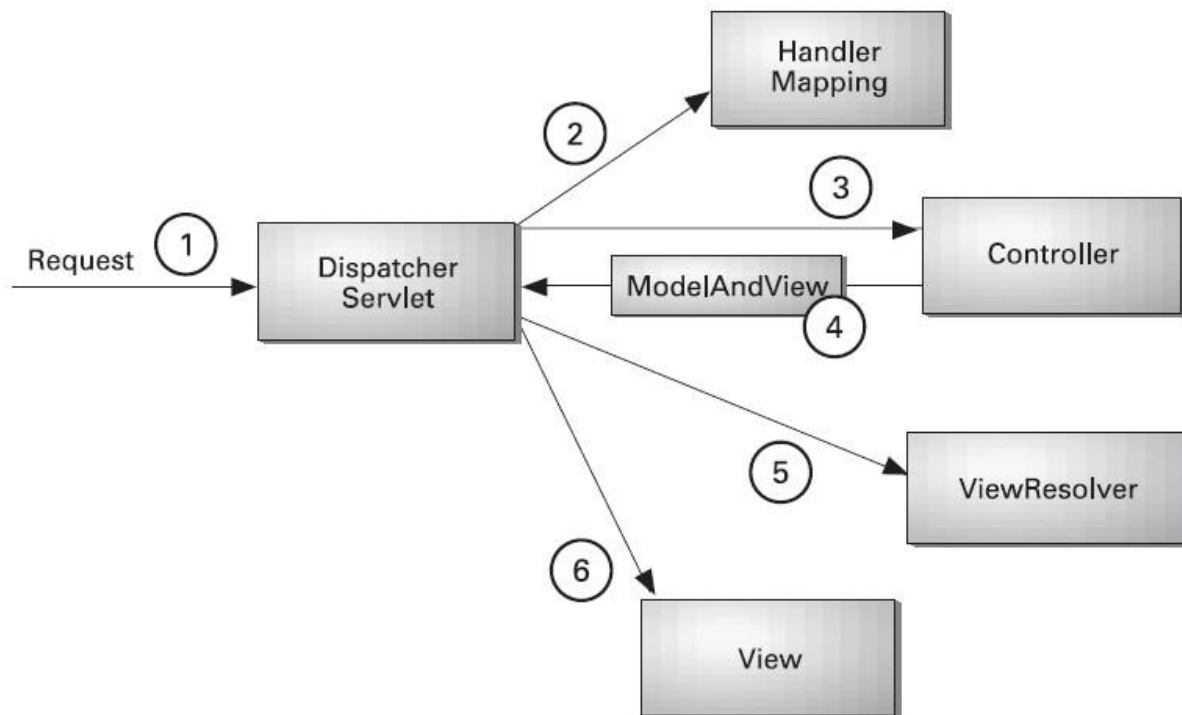
4.1 Use-Case Realizations

(n/a)

LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>

5. Logical View

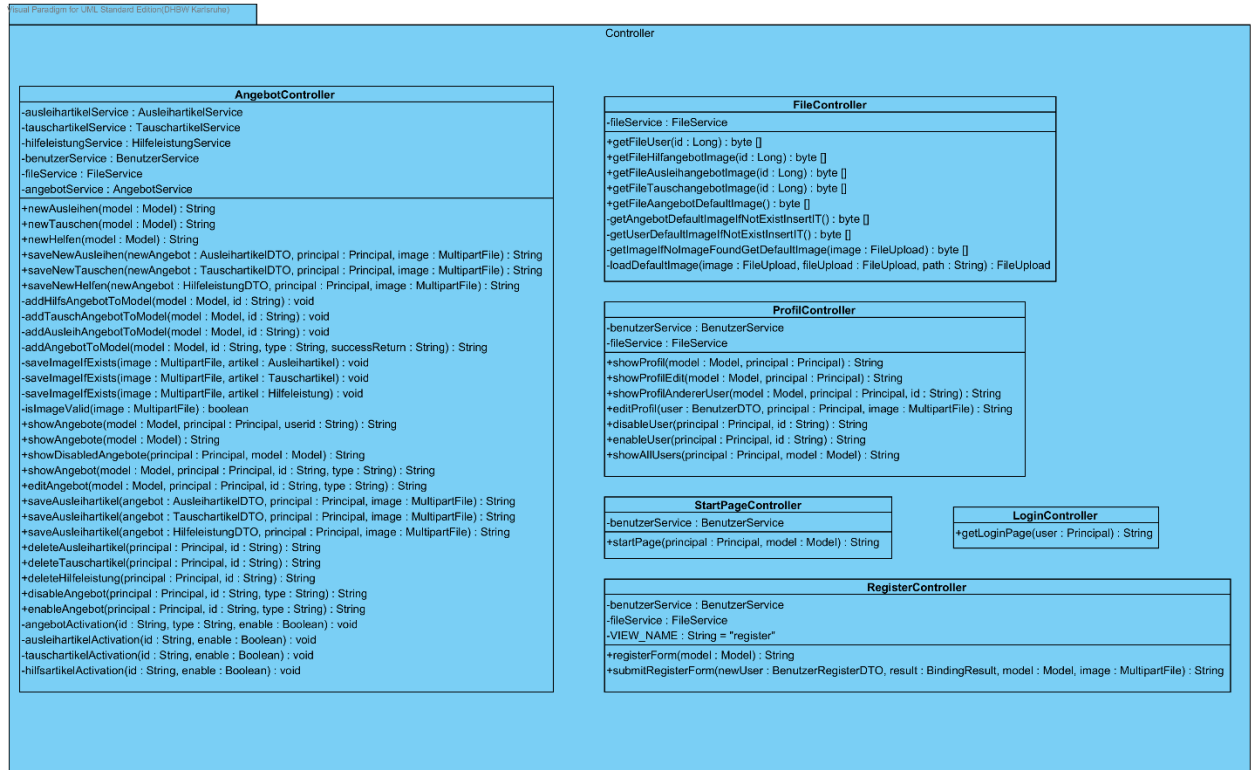
5.1 Overview



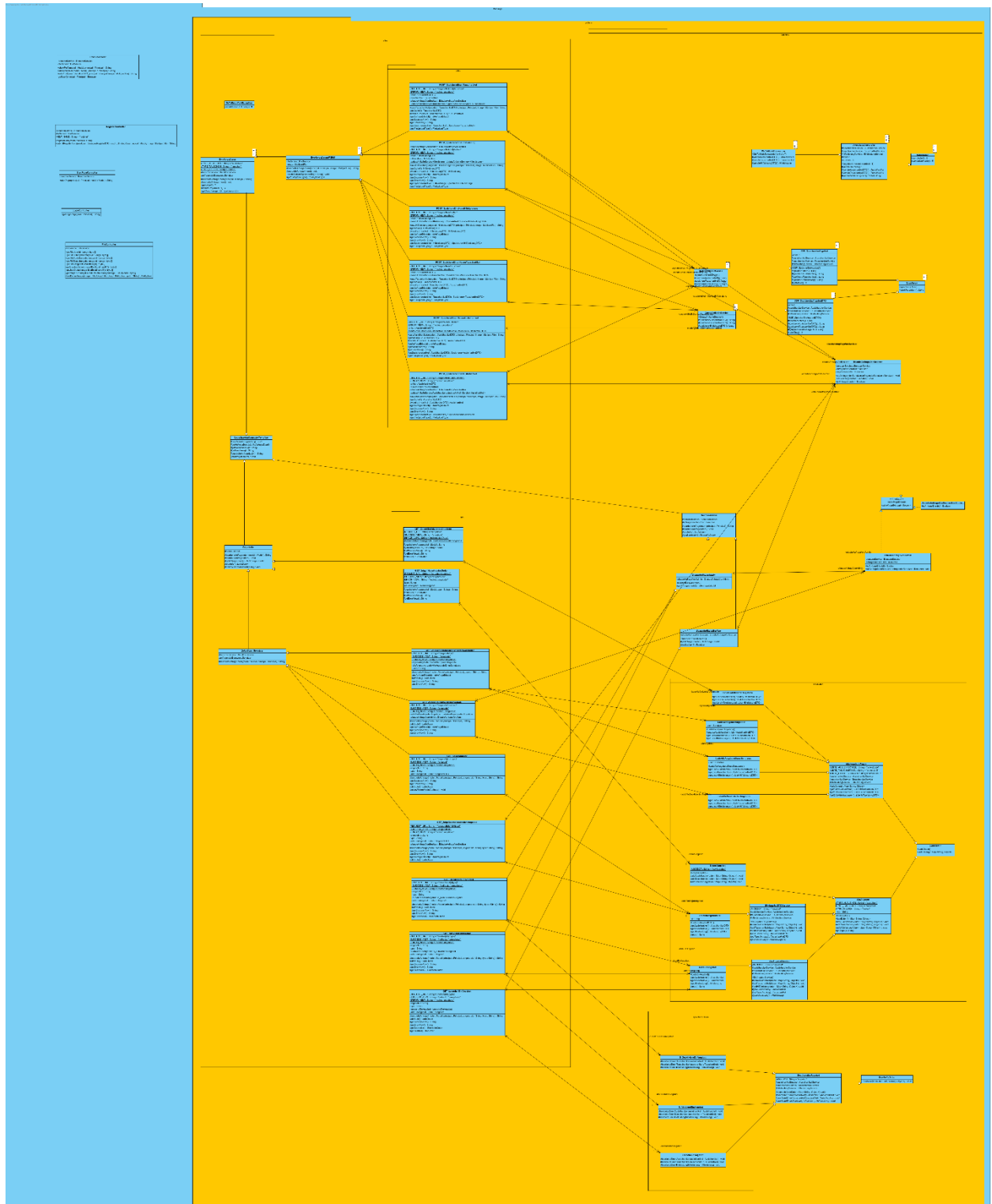
LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>

5.2 Architecturally Significant Design Packages

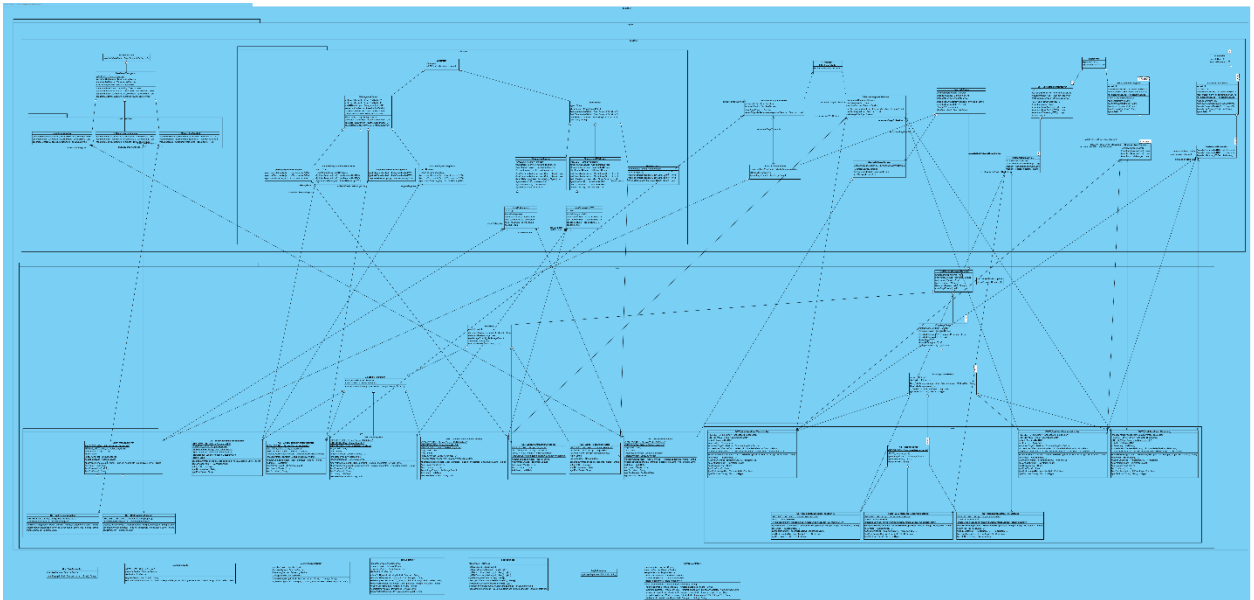
Vor Refactoring mit dem Template Method Pattern



Nach Refactoring mit dem Template Method Pattern



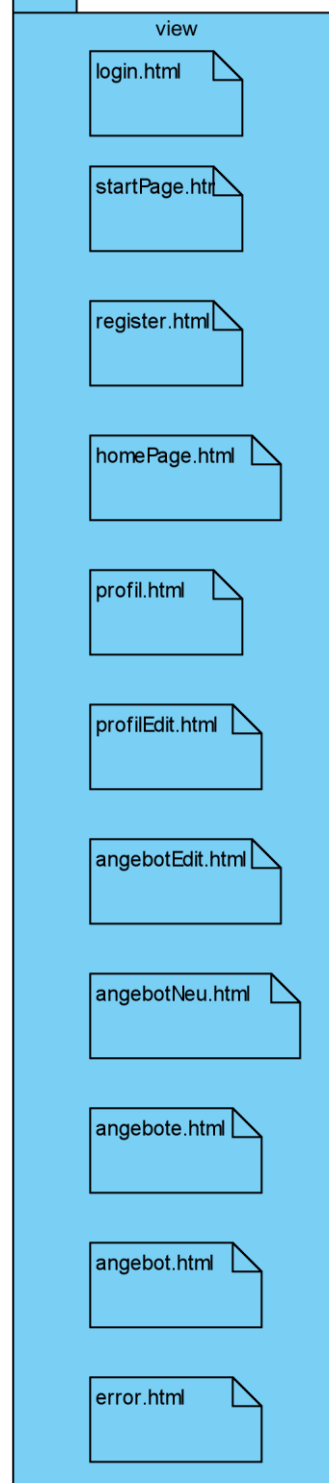
LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>



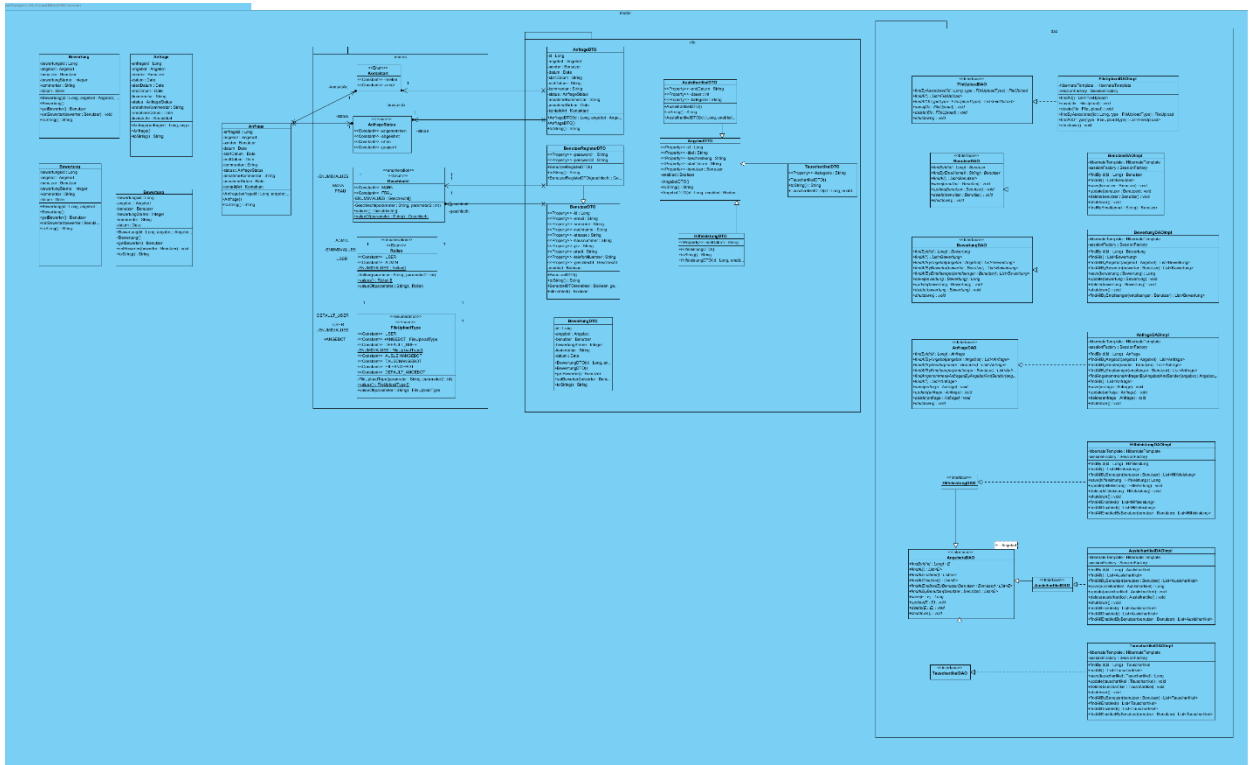
Controller

LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>

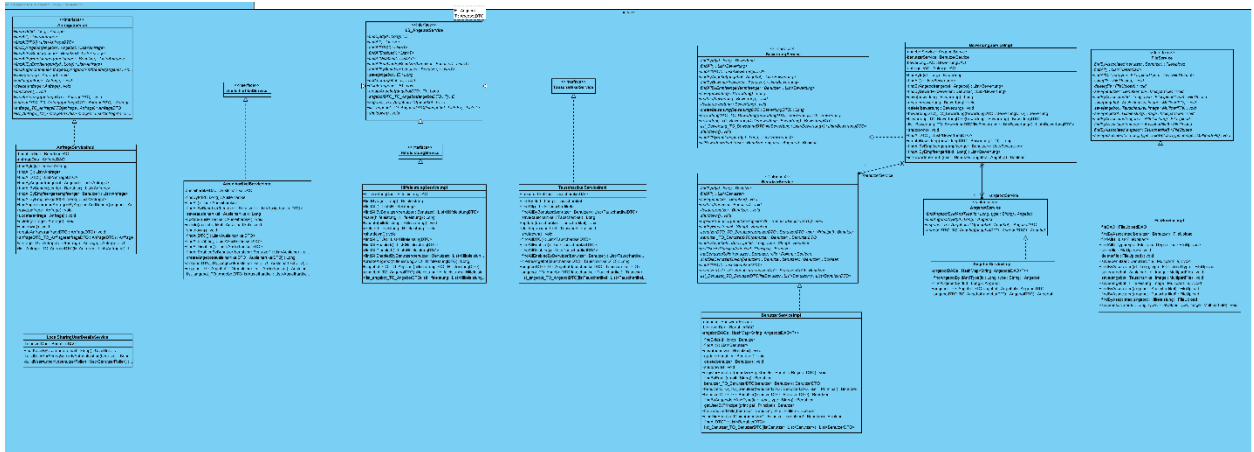
Visual Paradigm for UML Standard Edition(DHBW)



View

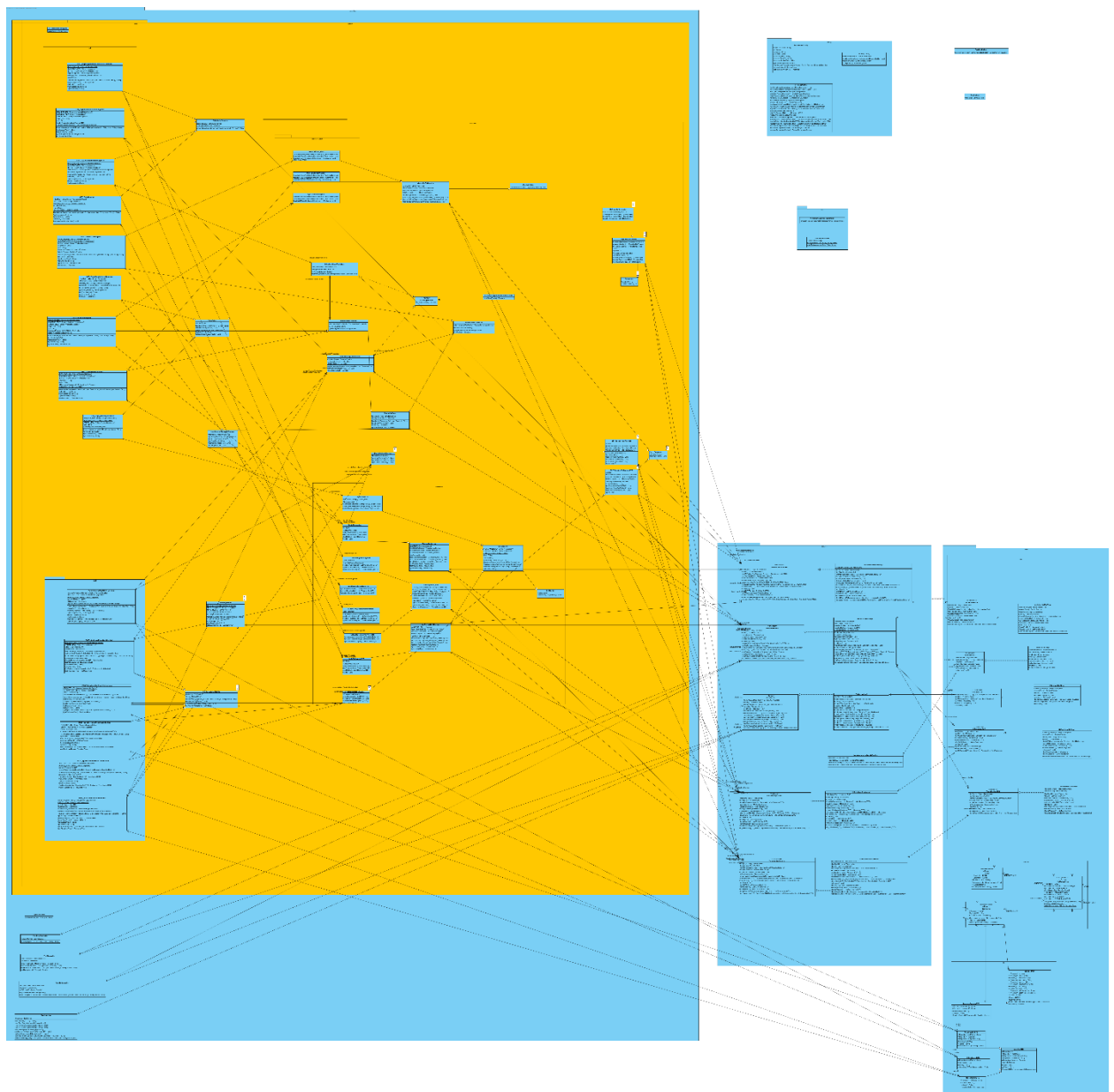


Model

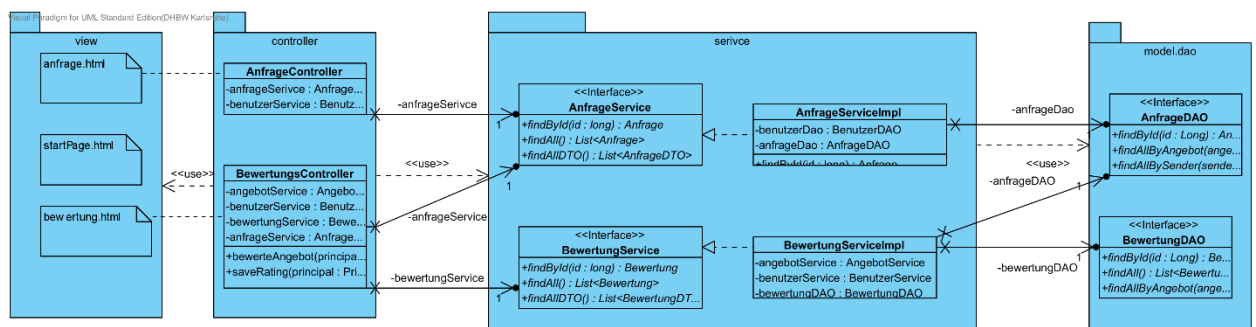


Service

LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>



LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>



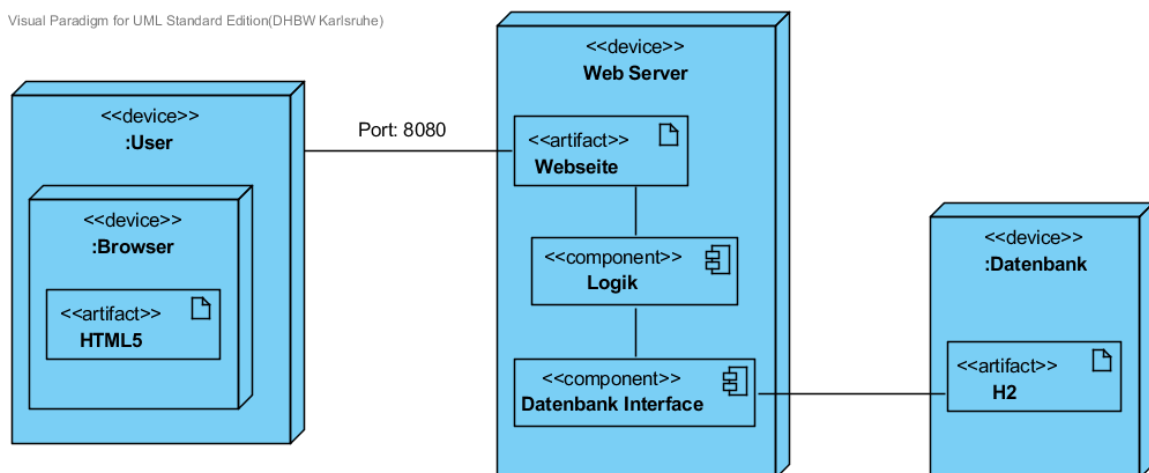
Übersicht

6. Process View

(n/a)

7. Deployment View

Visual Paradigm for UML Standard Edition(DHBW Karlsruhe)



8. Implementation View

(n/a)

8.1 Overview

(n/a)

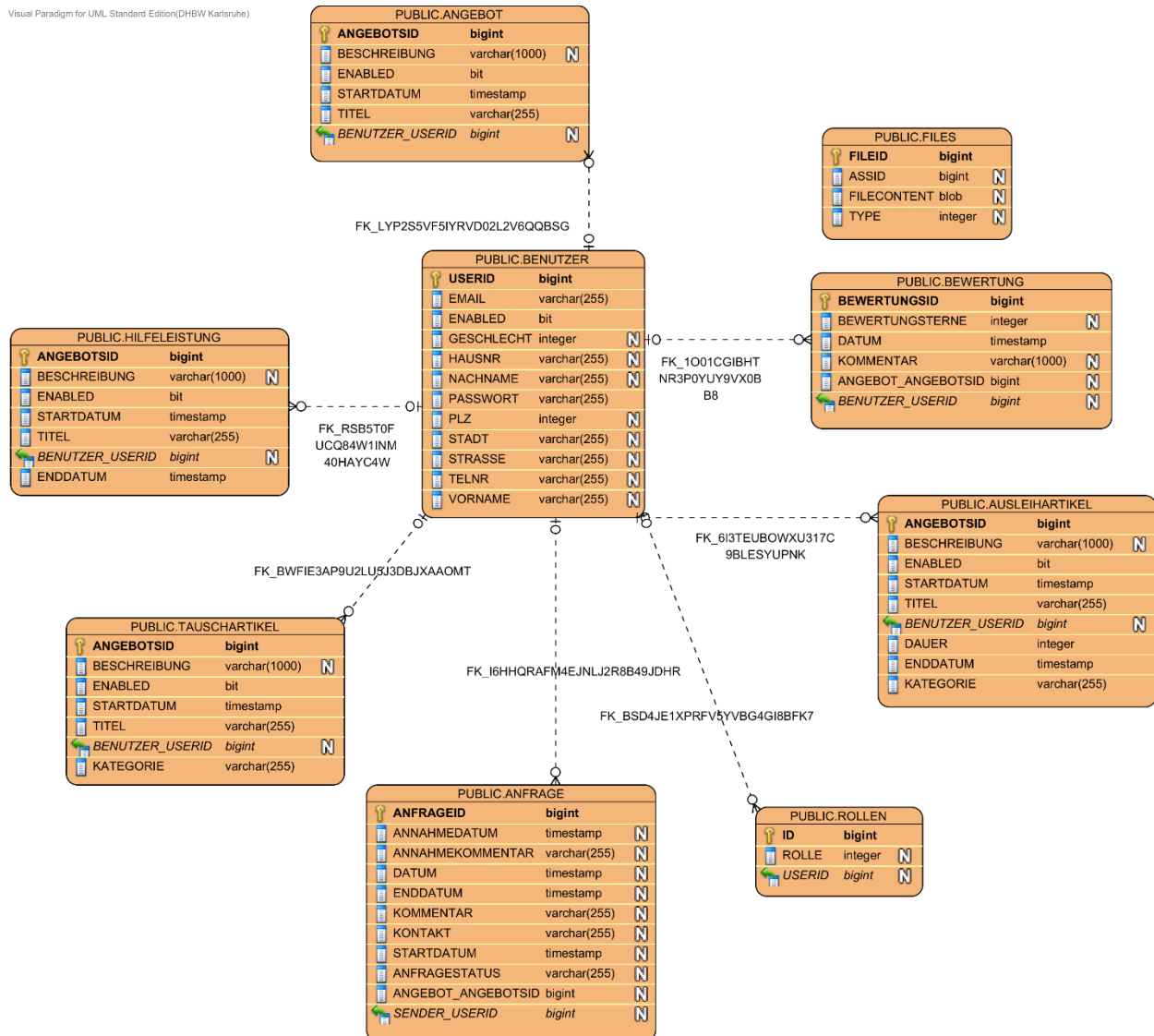
8.2 Layers

(n/a)

LocalSharing	Version: 1.0
Software Architecture Document	Date: <01/Jul/2015>

9. Data View (optional)

Visual Paradigm for UML Standard Edition(DHBW Karlsruhe)



10. Size and Performance

(n/a)

11. Quality

(n/a)