**Dokumentation**

**Entwicklung einer Webanwendung**

**für ein Ticketsystem**

**in der Fiducia & GAD IT AG**

Dokumentation für das 6. Semester

Abgabe am 31.03.2017

David Feldhoff

Moorstraße 88a

48431 Rheine-Mesum

Leon Stapper

Buchdahlstr. 6

48429 Rheine

Hochschule Weserbergland

Studiengang: Wirtschaftsinformatik

Studiengruppe: WI 44/14

Betreuender Dozent: Herr Malte Wildt

1. Inhaltsverzeichnis

[I Inhaltsverzeichnis I](#_Toc476910452)

[II Abkürzungsverzeichnis III](#_Toc476910453)

[III Abbildungsverzeichnis IV](#_Toc476910454)

[IV Tabellenverzeichnis IV](#_Toc476910455)

[1 Kurzbeschreibung der Anwendung 1](#_Toc476910456)

[2 Architekturbeschreibung 1](#_Toc476910457)

[2.1 Oberfläche 1](#_Toc476910458)

[2.2 Services 1](#_Toc476910459)

[2.3 Persistierung 1](#_Toc476910460)

[3 User Stories 1](#_Toc476910461)

[3.1 Anwender 1](#_Toc476910462)

[3.1.1 Veranstaltung suchen 1](#_Toc476910463)

[3.1.2 Details zur Veranstaltung einsehen 1](#_Toc476910464)

[3.1.3 Ticketreservierung 1](#_Toc476910465)

[3.1.4 Buchungscode zu einer Reservierung einsehen 1](#_Toc476910466)

[3.1.5 Einsicht der noch zur Verfügung stehenden Tickets 1](#_Toc476910467)

[3.2 Manager 1](#_Toc476910468)

[3.2.1 Login 1](#_Toc476910469)

[3.2.2 Veranstaltung anlegen 1](#_Toc476910470)

[3.2.3 Veranstaltung veröffentlichen 1](#_Toc476910471)

[3.2.4 Bearbeitung einer unveröffentlichten Veranstaltung 1](#_Toc476910472)

[3.2.5 Einsicht aller Reservierungen 1](#_Toc476910473)

[4 Sprint Backlog 1](#_Toc476910474)

[5 Darstellung des Datenmodells 1](#_Toc476910475)

[5.1 Klassendiagramm 1](#_Toc476910476)

[V Anhangsverzeichnis V](#_Toc476910417)

[VI Anhang A-1](#_Toc476910418)

1. Abkürzungsverzeichnis

|  |  |
| --- | --- |
| **Abkürzung** | **Bedeutung** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Abbildungsverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

[Abbildung 3: Mockup\_Widgetkatalog A-1](#_Toc476910419)

1. Tabellenverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

# Kurzbeschreibung der Anwendung

# Architekturbeschreibung

## Allgemeines

Die Anwendung "eventon" wurde als Model-2-Architektur realisiert. Daher unterteilt sich die Anwendung in mehrere Schichten. Die erste Schicht, bestehend aus xhtml-Dateien, realisiert die grafische Darstellung. Die zweite Schicht besteht aus .java-Klassen und bildet die Logik der Anwendung ab. Sie nimmt die Anfragen entgegen, verarbeitet sie und leitet sie an die Persistierungsschicht weiter. Die Persistierungsschicht besteht ebenfalls aus .java-Klassen, die den Zugriff auf die Datenbank realisieren.

## xhtml-Dateien für die Oberfläche

Die Oberfläche wurde ausschließlich in den xhtml-Dateien realisiert. Dabei basieren alle Seiten auf der Datei base.xhtml. Diese beinhaltet den grundlegenden Aufbau der Seite, bestehend aus einer Navigationsbar und dem Seiteninhalt. Dementsprechend wurde dient die base-Seite als grundlegende Struktur für das Templating. Zur Gestaltung der Oberfläche wurde das CSS-Framework "bootstrap" eingesetzt. Dieses wurde ebenfalls über die Templating-Seite eingebunden, um eine Verfügbarkeit über die gesamte Anwendung hinweg zu gewährleisten ohne auf jeder Seite "bootstrap" erneut einbinden zu müssen.

Bei der Entwicklung der Oberflächen-Dateien wurde strikt darauf geachtet, dass die xhtml-Dateien ausschließlich mit den vorgesehenen Logik-Klassen kommunizieren. Eine Kommunikation mit den Service-Klassen der Persistierungsschicht wurde bewusst vermieden, um die Verantwortlichkeiten der einzelnen Klassen strikt zu trennen.

Das Databinding zwischen xhtml-Dateien und den javaseitigen Logikklassen erfolgt über die Namensgebung der abgebildeten Attribute.

## Java-Forms zur Logik

Die Logik besteht aus javaseitigen Klassen. Der Name dieser Klassen endet immer mit "Form".

Für das beschriebene Databinding zwischen den Form-Klassen und den xhtml-Dateien wurden ausschließlich einzelne Attribute genutzt. Für die Bearbeitung von Attributen eines Objektes wurden zunächst alle Objektattribute in die Form-Attribute übertragen. Bei Abschluss der Bearbeitung wurden die Form-Attribute ins Objekt übernommen. Durch diesen Umweg sollte gewährleistet werden, dass Objekte ausschließlich aus der Form bearbeitet werden können.

Falls eine Konvertierung oder Validierung des einzelnen Eingabewertes notwendig war, erfolgte dies mit Hilfe von Convertern und Validatoren, die über entsprechende jsf-Attribute bzw. jsf-Tags an die Eingabewidgets angebunden wurden. Validierungen des gesamten erstellten Objektes erfolgen wiederum in der Logik, da die Validatoren lediglich die Eingabewerte einzelner Felder übergeben bekommen. Eine Ermittlung sämtlicher Eingabewerte wurde als zu umständlich angesehen.

Bei Abschluss einer Aufgabe, die Werte verändert, gelöscht oder neu erstellt hat, wurden aus den Forms entsprechende Methoden aus der Persistierungsschicht aufgerufen. Die Services der Persistierungsschicht sind dabei mit Interfaces in den Form-Klassen hinterlegt und werden mittels Dependency-Injection angezogen.

## Service-Interfaces

Wie bereits im vorherigen Kapitel beschrieben, ist den Service-Klassen der Persistierungsschicht ein Interface vorgeschaltet, sodass nie direkt mit den Service-Klassen kommuniziert wird, sondern lediglich deren Schnittstelle verwendet wird. Durch die dadurch entstehende Abstraktion sollte eine möglichst lose Kopplung zwischen Logik und Datenhaltung gewährleistet werden.

## Services zur Datenhaltung und Persistierung

Die Services zur Datenhaltung und Persistierung implementieren die zuvor beschriebenen Interfaces. Die Services beziehen über Dependency-Injection eine Instanz des EntityManagers zur Kommunikation mit der Datenbank. Darüber können Objekte in der Datenbank erstellt, bearbeitet und gelöscht werden. Außerdem können hierüber Objekte ausgelesen werden. Dafür wurden unter anderem auch kriterienbasierte Suchanfragen mit Hilfe des "CriteriaBuilders" erstellt.

## Session-Informationen über den SessionContext

Zusätzlich zu den Informationen, die in der

# User Stories

## Anwender

### Veranstaltung suchen

### Details zur Veranstaltung einsehen

### Ticketreservierung

### Buchungscode zu einer Reservierung einsehen

### Einsicht der noch zur Verfügung stehenden Tickets

## Manager

### Login

### Veranstaltung anlegen

### Veranstaltung veröffentlichen

### Bearbeitung einer unveröffentlichten Veranstaltung

### Einsicht aller Reservierungen

# Sprint Backlog

|  |  |
| --- | --- |
| Aufgabe | Bearbeiter |
| Projekt aufsetzen (gradle, bootstrap) | Leon Stapper |
| Mockups erstellen | Leon Stapper |
| Templating | Leon Stapper |
| Navigation | Leon Stapper |
| EventService zur Event-Verwaltung | Leon Stapper |
| UserService zur User-Verwaltung | Leon Stapper |
| Registrierung | David Feldhoff |
| Login | Leon Stapper |
| Events erstellen und bearbeiten | Leon Stapper |
| Bilderupload | David Feldhoff |
| Verwaltung unveröffentlichte Events | David Feldhoff |
| Verwaltung veröffentlichte Events | David Feldhoff |
| Event-Suche | Leon Stapper |
| Event anzeigen buchen | Leon Stapper |
| User-Übersicht | David Feldhoff |
| Regex-Entwicklung zur Validierung | David Feldhoff |
| ER-Modell erstellen | David Feldhoff |
|  |  |
|  |  |
|  |  |

# Darstellung des Datenmodells

## Klassendiagramm

Anhangsverzeichnis

[A1 Mockups A-1](#_Toc476910383)

A

Anhang

1. Mockups

Im Folgenden finden sich die Oberflächenentwürfe bzw. Mockups zum Ticketsystem wieder.

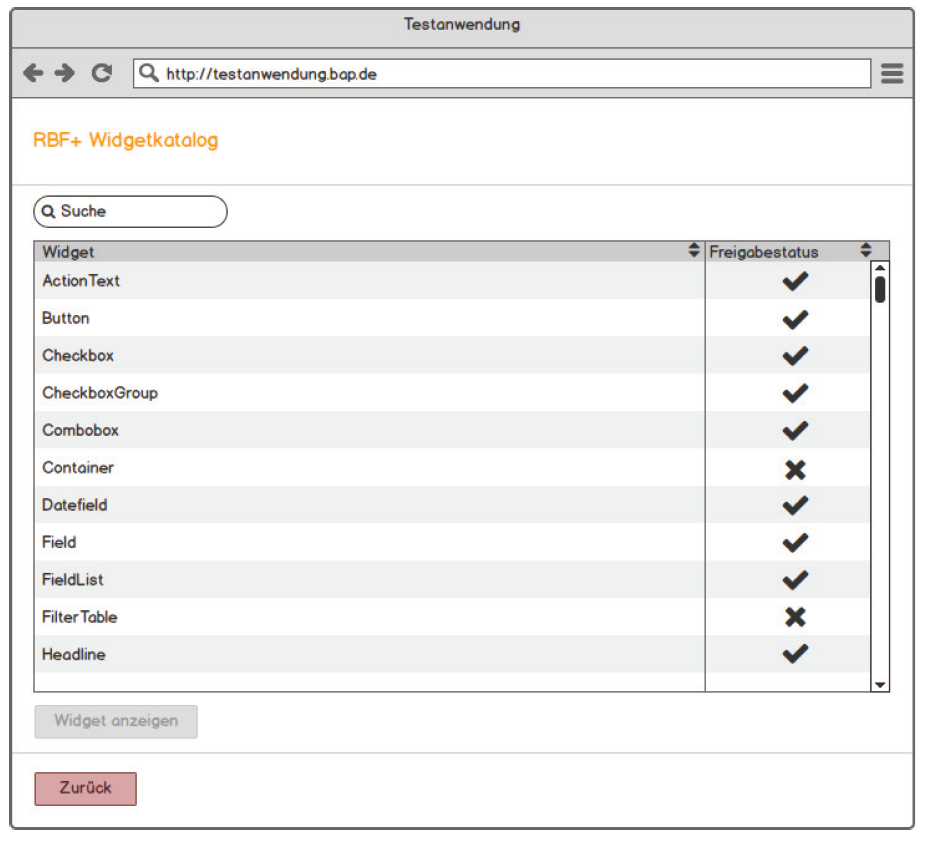


Abbildung : Mockup\_Widgetkatalog

**Eigenständigkeitserklärung**

"Wir versichern hiermit, dass wir die Arbeit selbstständig verfasst, keine anderen Quellen und Hilfsmittel als die angegebenen benutzt und die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht haben. Das Gleiche gilt auch für eingefügte Zeichnungen, Kartenskizzen und Darstellungen.”

|  |  |
| --- | --- |
| Hameln, den 29.03.2017 |  |
| Ort, Datum | Unterschrift |

|  |  |
| --- | --- |
| Hameln, den 29.03.2017 |  |
| Ort, Datum | Unterschrift |