****

**Internationaler Studiengang Medieninformatik**

**Datenbankbasierte Web-Anwendungen (SoSe 2015)**

**Autovermietung**

|  |  |
| --- | --- |
| Fabian Redecker | 375750 |
| Marco Geils | 375736 |
| Mike Hüsing | 344089 |
| Oliver Bammann | 360330 |
| Konrad Mumm |  |

Inhaltsverzeichnis

1. Spezifikation 3

1.1 Anforderungen 3

2. Architektur 3

2.1 Gesamtarchitektur 3

2.2 Programmabläufe 3

2.3 Persistenz 3

2.3.1 Konzeption 3

2.3.2 Umsetzung 3

2.3.3 Zugriff 4

2.4 Präsentation 4

3. Installation 4

3.1 Programmierwerkzeuge 4

3.1.1 Entwicklungsumgebung: *Intellij* 4

3.1.2 Versionskontrollsystem: *Git* 4

3.1.3 Buildautomatisierungswerkzeug: *Gradle* 4

3.1.4 Komponententest-Framework: 4

3.2 Installation der Entwicklungsumgebung 4

3.3 Installation der Anwendung 4

3.4 Konfiguration der Datenbank 5

# 1. Spezifikation

*In diesem Kapitel sollen die folgenden Punkte erläutert werden:*

* *Was leistet die Webanwendung?*

Mit der Webanwendung „Autovermietung“ lassen sich Daten einer potentiellen Autovermietungsfirma einsehen, bearbeiten oder hinzufügen. Dabei werden die Daten in einer Datenbank gelesen bzw. geschrieben. Bevor über einen Browser auf die Daten zugegriffen werden kann, erfolgt zunächst ein Login mit Name/E-Mail und Passwort.

* *Welchen Mehrwert bietet die Webanwendung für dessen Nutzer?*

## 1.1 Anforderungen

*In diesem Unterkapitel sollen die vorgesehenen und die tatsächlich umgesetzten Anforderungen beschrieben und mit Hilfe von Use-Case- und Aktivitätsdiagrammen konkretisiert werden.*

**Vorgesehene Anforderungen:**

„Als Mitarbeiter möchte ich Fahrzeuge in die Datenbank eintragen und löschen können, um den Datenbankbestand der Fahrzeuge aktualisieren zu können.“

„Als Mitarbeiter möchte ich für die Autovermietung die Kundenliste einsehen und bearbeiten können, um neue Kunden eintragen zu können und Informationen über diese zu erhalten.“

„Als Kunde möchte ich Fahrzeuge buchen können, um mobil zu sein.“

„Als Kunde möchte ich abrufen können, welche Fahrzeuge nicht vermietet sind, um zu prüfen, ob das gewünschte Fahrzeug für meinen Termin verfügbar ist.“

„Als Kunde möchte ich sehen, welche Fahrzeuge im Fuhrpark vorhanden sind um zu sehen ob die Autovermietung mein gewünschtes Fahrzeug besitzt.“

# 2. Architektur

*In diesem Kapitel soll beschrieben werden, wie das Projekt und die Datenbank strukturiert ist und welche Technologien und Frameworks verwendet werden.*

## 2.1 Gesamtarchitektur

*In diesem Unterkapitel soll beschrieben werden, wie die Gesamtarchitektur der entwickelten Webanwendung aussieht. Dies kann mit UML Paket- oder Klassendiagrammen unterstützt werden.*

Das Projekt ist in einer Drei-Schichten-Architektur organisiert. Die Schichten sind Datenschicht, Logikschicht und Präsentationsschicht.

* Datenschicht

In der Datenschicht finden mittels Hibernate(OR-Mapping) zugriffe auf die MySql Datenbank statt.

* Logikschicht

Hier befinden sich die DAOs(Data Access Object). Diese generieren mittels querry statements die CRUD(Create Read Update Delete) Funktionen.

* Präsentationsschicht

Hier sind die JSPs (Java Server Pages) zu finden. Die JSPs beinhalten HTML Code gepaart mit Java Code. In diesem Code werden die DAOs aus der Logikschicht abgerufen.

Die Präsentationsschicht nimmt HTTP-Requests entgegen. Dieser Request wird dann von der Logikschicht verarbeitet und Daten aus der Datenschicht abgerufen. Die Java Server Pages, die in der Präsentationschicht befinden generieren ein Servlet welches HTML ausliefert. Somit gibt die Präsentationschicht HTTP-Responses zurück.

## 2.2 Programmabläufe

*In diesem Unterkapitel sollte exemplarisch gezeigt werden, wie die typischen Abläufe in der Webanwendung aussehen. Hierfür sollten Aktivitäts-/Sequenzdiagramme genutzt werden.*

## 2.3 Persistenz

*In diesem Unterkapitel soll die genutzte Datenbank und der Zugriff auf ihre Daten erläutert werden.*

### 2.3.1 Konzeption

*In diesem Unterkapitel soll die konzepierte Datenbank nach ER-Diagramm beschrieben werden*

### 2.3.2 Umsetzung

*In diesem Unterkapitel soll die Datenbank nach relationalem Modell beschrieben werden. Außerdem sollten anwendungsspezifische Constraints aufgelistet werden, die bei der Implementierung bedacht werden müssen.*

### 2.3.3 Zugriff

*In diesem Unterkapitel soll beschrieben werden mit welchen Technologien und Frameworks von der Webanwendung auf die Datenbank zugegriffen wird.*

## 2.4 Präsentation

*In diesem Unterkapitel soll die Präsentationsschicht der Webanwendung beschrieben werden. Hierzu u.a. die genutzte Technologie/Framework.*

# 3. Installation

*In diesem Kapitel soll beschrieben werden, welche Programmierwerkzeuge verwendet wurden und wie die Webanwendung lokal aufgesetzt werden kann*

## 3.1 Programmierwerkzeuge

*Auflistung der verwendeten Programmierwerkzeugen mit Vor- und Nachteilen*

### 3.1.1 Entwicklungsumgebung: *Intellij*

Der Hauptgrund, dass wir IntelliJ verwendeten, war, dass eine Person bei uns Erfahrung damit hatte und in vorherigen Projekten die Einfachheit des Einbindens und Benutzens von Git kennengelernt hatte. Ein weiterer Vorteil von IntelliJ ist die, wie der Name schon verrät, Eigenintelligenz des Programmes. Autovorschläge und Code-Generierung sind sehr gut in der Entwicklungsumgebung. Negativ aufgefallen ist uns allerdings das Fehler Highlighting und das teils unübersichtliche Design.

### 3.1.2 Versionskontrollsystem: *Git*

Sowie bei IntelliJ war der Hauptgrund, dass wir Git benutzten unsere angesammelte Erfahrungen mit Git, die sogar noch mehr sind, als die von IntelliJ. Uns wurde früh im Studium nahegelegt Git zu verwenden, da es sehr gut im Mergen ist und wir keinen eigenen Server benötigen, wie für z.B. SVN. Ein Nachteil von Git ist, dass es sehr kompliziert ist zu lernen und auch nur wenig gute GUI’s hat.

### 3.1.3 Buildautomatisierungswerkzeug: *Gradle*

Nachdem wir uns über die Buildtools erkundigt hatten, haben wir uns für Gradle entschieden, da es die Vorteile von Maven und Ant bietet und dabei sehr einfach zu installieren ist. Diese Vorteile sind auch teils dadurch, dass Gradle das modernste der drei Systeme ist und von den anderen Tools lernen konnte.

### 

### 3.1.4 Komponententest-Framework:

Wir haben uns gegen die Nutzung eines Komponententest-Frameworks entschieden.

Aus unserer Sicht haben wir uns in dem Modul mit vielen neuen Sachen beschäftigt. Wir wissen das „Test-First“ ein wichtiger Aspekt der Agilen Programmierung ist. Allerdings haben wir uns im letzten Semester nur kurz und theoretisch mit dem Thema beschäftigt.

## 3.2 Installation der Entwicklungsumgebung

Zum Einrichten der Entwicklungsumgebung gehört das Installieren von Gradle, IntelliJ und vor allem einer aktuellen Java Runtime dazu. Um dann das Projekt einzubinden muss Git installiert und eingerichtet werden. Das Projekt kann nun in IntelliJ eingebunden werden.

## 3.3 Installation der Anwendung

*Beschreibung wie die Anwendung auf dem lokalen Rechner gestartet werden kann*

## 3.4 Konfiguration der Datenbank

*Beschreibung welche Schritte durchgeführt werden müssen, um die Datenbank mitsamt ihrer Struktur und Beispieldaten auf einem lokalen System aufzusetzen*