

Semesteraufgabe: Java EE Architektur SS 2017

Kurzbeschreibung der Anwendung

Die entwickelte Softwarelösung orientiert sich an der Model 2-Architektur. Die Komponenten der Anwendung lassen sich demnach unterschiedlichen Schichten zuordnen.

Die erste Schicht behandelt die Darstellung der Inhalte und wird daher im folgenden als Präsentationsschicht bezeichnet. Zu ihr zählen die Views welche im Verzeichnis `src/main/webapp/` liegen.

Die Schicht der Geschäftslogik hingegen beschäftigt sich mit der konkreten Implementierung jeglicher Funktionalitäten zur Realisierung der User-Stories. Zu ihr zählen daher die Controller- und Request-Komponenten.

Die letzte Schicht beschäftigt sich mit der Speicherung der Daten und wird daher als Persistenzschicht bezeichnet. Zu ihr zählen die Datenmodelle und die Service-Komponenten, welche die Historisierung in H2 realisieren.

Tabelle 1: Architektur der Anwendung

Anwendungsschicht	Beschreibung
Präsentation	Views für die Abbildung der User-Stories
Geschäftslogik	Steuerung der Präsentation und Abbildung von Funktionalitäten
Persistenz	Datenmodell und Historisierung in H2

Umsetzung der User-Stories

Liste aller beteiligten Komponenten und Klassen, sowie deren Aufgaben und Zugehörigkeit der Anwendungsschicht.

Kurze Beschreibung der Schritte, die ein Nutzer für die Anwendung der Story ausführen muss.

1. Veranstaltung anlegen

Komponente	Aufgabe	Anwendungsschicht
<code>createEvent.xhtml</code>	Darstellung des Formulars	Präsentation
<code>CreateEventRequest.java</code>	Steuerung des Formulars	Geschäftslogik
<code>SessionContext.java</code>	Authentifizierung	Geschäftslogik
<code>ManagementOperation.java</code>	Autorisierung: Ist der angemeldete User ein Manager?	Geschäftslogik

EventService.java	Historisierung	Persistenz
Event.java	Datenmodell	Persistenz

Der Manager wählt nach dem Login zunächst im linken Navigationsbereich den Eintrag **Veranstaltung anlegen** aus. Danach erscheint ein Formular in dem die nötigen Angaben zum Erstellen einer Veranstaltung abgefragt werden. Ist der Manager fertig mit der Eingabe der benötigten Daten, kann die Veranstaltung über den gleichnamigen Button erstellt werden.

2. Veranstaltung veröffentlichen

Komponente	Aufgabe	Anwendungsschicht
publishEvents.xhtml	Auflistung seiner noch nicht veröffentlichten Veranstaltungen	Präsentation
SessionContext.java	Authentifizierung	Geschäftslogik
ManagementOperation.java	Autorisierung: Ist der angemeldete User ein Manager?	Geschäftslogik
EventService.java	Durchführen der Veröffentlichung inklusive Historisierung	Persistenz
Event.java	Datenmodell	Persistenz

Der Manager wählt nach dem Login zunächst im linken Navigationsbereich den Eintrag **Veranstaltung veröffentlichen** aus. Danach erscheint eine Liste der noch nicht veröffentlichten Veranstaltungen. Hier hat der Manager die Option eine Veranstaltung aus der Liste direkt zu veröffentlichen. Alternativ hierzu kann er sich auch zunächst die Details anzeigen lassen um vorher noch einmal die eingegebenen Daten näher zu kontrollieren und ggf. zu bearbeiten.

3. Veranstaltung bearbeiten

Komponente	Aufgabe	Anwendungsschicht
changeEvent.xhtml	Darstellung des Formulars	Präsentation
ProcessEvent.java	Steuerung des Formulars	Geschäftslogik
SessionContext.java	Authentifizierung	Geschäftslogik
SecurityContext.java	Autorisierung: Ist der angemeldete User der Ersteller der Veranstaltung?	Geschäftslogik
EventService.java	Historisierung	Persistenz
Event.java	Datenmodell	Persistenz

Der Manager wählt nach dem Login zunächst im linken Navigationsbereich den Eintrag **Meine Veranstaltungen** aus. Aus der Liste der Veranstaltungen wählt (**Details**) er dann eine aus, die er gerne bearbeiten möchte. Hier bekommt der Manager zunächst eine Übersicht über die aktuellen Eigenschaften der Veranstaltung und hat die Option, die **Veranstaltung zu bearbeiten**. Die Anpassungen bestätigt der Manager mit einem Klick auf den Button **Änderungen speichern**, dadurch wird er zurück auf die Übersicht seiner Veranstaltungen geleitet.

4. Veranstaltung suchen

Komponente	Aufgabe	Anwendungsschicht
search.xhtml	Darstellung des Formulars	Präsentation
SearchRequest.java	Steuerung des Formulars	Geschäftslogik
SessionContext.java	Authentifizierung	Geschäftslogik
EventService.java	Durchführen der Abfrage	Persistenz
Event.java	Datenmodell	Persistenz

Der Anwender wählt nach dem Login zunächst im oberen Navigationsbereich den Eintrag **Suche** aus. Dort gibt er einen Suchbegriff ein, dies kann beispielsweise der Ort **Berlin** sein. Nach der Bestätigung seiner Eingabe werden ihm alle Veranstaltungen die zu seiner Suche passen aufgelistet. Dabei wird der *Name*, die *Beschreibung* und der *Ort* der Veranstaltung berücksichtigt.

Alternative: Darüber hinaus kann der Anwender bei jeder Auflistung von Veranstaltungen über die Filter unter den Spaltenüberschriften nach spezifischen Veranstaltungen suchen. Hierzu werden Filter für den *Namen*, die *Art* und den *Ort* der Veranstaltung angeboten.

5. Veranstaltung ansehen

Komponente	Aufgabe	Anwendungsschicht
events.xhtml	Auflistung der Veranstaltungen	Präsentation
event.xhtml	Darstellung der Details	Präsentation
SessionContext.java	Authentifizierung	Geschäftslogik
EventService.java	Durchführen der Abfrage	Persistenz
Event.java	Datenmodell	Persistenz

Der Anwender erhält nach dem Login eine Übersicht der veröffentlichten Veranstaltungen, die in Zukunft stattfinden werden. Hierbei wird ihm *Name*, *Art*, *Ort*, *Datum* und das noch zur Verfügung stehende *Kontingent an Tickets* der Veranstaltung angezeigt. Nach einem Klick auf **Details** neben der Veranstaltung wird ihm zusätzlich eine weiterführende *Beschreibung* der Veranstaltung angezeigt.

6. Ticketreservierung

Komponente	Aufgabe	Anwendungsschicht
bookEvent.xhtml	Darstellung des Formulars	Präsentation
BookEventRequest.java	Steuerung des Formulars	Geschäftslogik
SessionContext.java	Authentifizierung	Geschäftslogik
ReservationService.java	Historisierung	Persistenz
Reservation.java	Datenmodell	Persistenz

Der Anwender hat nach dem Login eine Veranstaltung aus der Übersicht ausgewählt, die er gerne besuchen würde. In der Detail-Ansicht erhält er die Option **Ticket reservieren**. Hierbei wird ihm eingeblendet wieviele Karten aktuell noch zur Verfügung stehen. Nach einer Eingabe der gewünschten Anzahl an Tickets, kann der Anwender die **Reservierung bestätigen** oder den Vorgang **abbrechen**. Bestätigt er seinen Reservierungswunsch, so wird - wenn noch genügend Tickets zur Verfügung stehen - die Reservierung bestätigt.

7. Reservierungsbestätigung

8. Reservierungsübersicht

9. Noch reservierbare Tickets

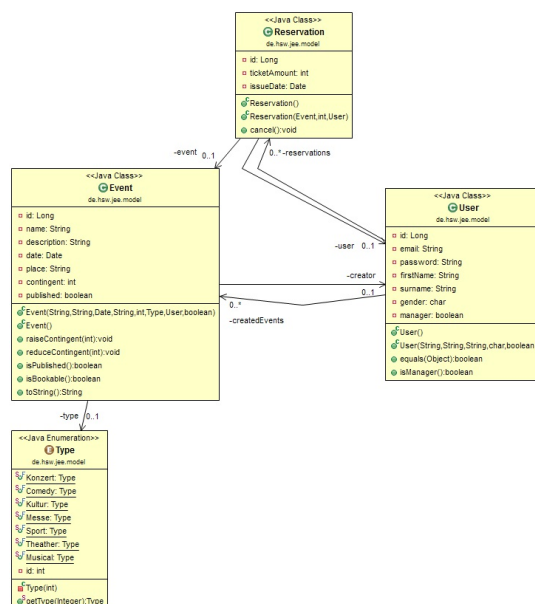
Tabelle 2: Sprint-Backlog

Task	Beschreibung	Story	Entwickler
Projekt-Setup	Git, Gradle, Eclipse, Primefaces, Templating	-	Florian
Datenmodell	Konzipierung der benötigten Entitäten	-	Gemeinsam
Authentifizierung	Registrierung & Anmeldung an der Anwendung	-	Arthur
Autorisierung	Zugriffsschutz für Seiten- und Methodenaufrufe, die Managern vorbehalten sein sollen	-	Florian
Veranstaltung anlegen		1	Arthur
Veranstaltung veröffentlichen	Veranstaltungen werden erst nach der Freigabe durch ihren jeweiligen Manager für andere sichtbar	2	Florian
Veranstaltung bearbeiten		3	Arthur
Veranstaltung suchen	Kontextbasiertes Durchsuchen zzgl. konkreter Filter in den Auflistungen	4	Gemeinsam

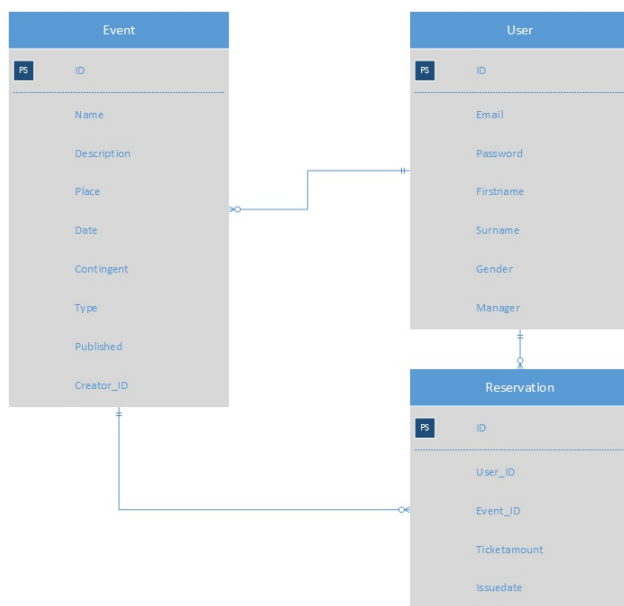
Veranstaltung ansehen		5 & 9	Arthur
Ticketreservierung	Reservieren von Tickets und Möglichkeit zur Stornierung	6 & 9	Florian
Reservierungsbestätigung	Vergabe und Mitteilen einer eindeutigen ID für jede Reservierung	7	Florian
Auflistung von Reservierungen	Auflistung aller Reservierungen zu den Veranstaltungen des angemeldeten Managers	8	Florian

Datenmodell

Klassendiagramm



Beschreibung der Datenbankstruktur



Die Datenbankstruktur wird bei deployment der Webanwendung automatisch erstellt. Die einzelnen Datenbanktabellen werden von der Java Persistence API anhand der Klassen die als Entity gekennzeichnet sind abgeleitet und automatisch generiert. Diese Persistence API übernimmt zusätzlich zum ableiten der Datenbankstruktur noch einige weitere Aufgaben. So sorgt sie auch für das Erstellen einer eindeutigen ID für die jeweiligen Instanzen der Klassen, kümmert sich um die Integrität der Datenbank und das Verwalten der Beziehungen zwischen den einzelnen Entitäten.

Insgesamt sind drei Entity für die Webanwendung entscheidend:

- Zunächst wäre hier der User zu nennen. Er repräsentiert einen realen Nutzer der Webanwendung und wird mit der Kombination aus E-Mail-Adresse und einem Passwort gekennzeichnet. Ein User kann zusätzlich als Manager markiert werden. Dies gibt ihm die Möglichkeit Veranstaltungen zu erstellen, diese frei zu geben, zu bearbeiten und zugehörige Reservierungen einzusehen.
- Eine weitere Entität wäre das Event. Sie stellt eine Veranstaltung da, bestehend aus einem Namen, einer Beschreibung, einem Veranstaltungsort, einem Datum mit Uhrzeit und einer Anzahl verfügbarer Karten. Zusätzlich kann jede Veranstaltung noch einer Kategorie zugeordnet werden. Eine Veranstaltung kann dabei entweder unveröffentlicht oder veröffentlicht sein. Eine unveröffentlichte Veranstaltung ist nur vom jeweiligen User der sie erstellt hat einsehbar und auch nur dieser kann sie veröffentlichen. Der Ersteller der Veranstaltung ist über die `creator_ID` gespeichert.
- Als letztes gibt es noch die Reservation. Sie bildet eine Reservierung für eine Veranstaltung von einem User ab. Als solche speichert sie diese auch mit Hilfe der Attribute `Event_ID` und `User_ID` ab. Zusätzlich besitzt sie noch das Attribut `ticketamount`, welches die Anzahl an reservierten Tickets abspeichert und `issuedate`, welches das Datum mit Uhrzeit speichert, an welchen die Reservierung erfolgte.