Use Case Diagramm

# Einleitung

Das Use Case Diagramm, oder auch als Anwendungsfalldiagramm bezeichnet, beschreibt alle möglichen Anwendungsfälle innerhalb einer Anwendung. Es besteht im Endeffekt lediglich aus Akteuren sowie Anwendungsfälle, welche mehr oder weniger detailliert dargestellt werden können. Es stellt somit das erwartete Verhalten eines Systems dar und wir verwendet um die Anforderungen an ein System zu spezifizieren.

# Akteure

## Hochzeitsplaner

Der Hochzeitsplaner ist der zentrale Akteur, da er praktisch die hauptverantwortliche Person für die Hochzeitsplanung ist. Er hat auf fast alles Zugriff, bis auf die Nutzerverwaltung, welche der Administrator inne hat.

## Unterhaltungsmanager

Der Akteur Unterhaltungsmanager kommt dann ins Spiel sobald es um Aktionen geht. Er kann solche anlegen und diese ebenfalls verwalten.

## Administrator

Der Akteur Administrator ist lediglich für die Personenverwaltung relevant, da dies seine Hauptaufgabe, neben der generellen Systemadministration, ist. Allerdings verfügt er über eine Sonderstellung, da er als Administrator natürlich über alle Rechte verfügt und somit auch Zugriff auf alles hat.

## Benutzer

Der Akteur Benutzer ist der Basisakteurtyp, das heißt: der Hochzeitsplaner, der Untehaltungsmanager, sowie der Administrator sind ebenfalls Benutzer, haben jedoch zusätzliche Berechtigungen. Dieser Akteur dient somit als Verallgemeinerung für Anwendungsfälle die theoretisch von jedem Benutzer ausgeführt werden kann.

# Use Cases

## Hochzeitsveranstaltung managen

Der Anwendungsfalls Hochzeitsveranstaltung managen ist der „Haupt“-Use Case, denn er beinhaltet fast alle weiteren Use Cases der Anwendung. Ebenfalls ist die Verwaltung auch der Hochzeit auch der Hauptaspekt des Programmes. In diesem Anwendungsfall sind drei Akteure vertreten: Hochzeitsplaner, Unterhaltungsmanager und Benutzer.

Der Hochzeitsplaner hat eine zentrale Rolle in diesem Diagramm, denn er ist für die Verwaltung zuständig und kann somit eine Hochzeitsveranstaltung anlegen, anzeigen, sowie verwalten. „Hochzeitsveranstaltung anlegen“ ist ein Fall der lediglich durch den Hochzeitsplaner selbst ausgeführt wird. Der Anwendungsfall Hochzeitsveranstaltung verwalten enthält weitere Use Cases, dazu zählen: Hilfsmittel verwalten, Aktion verwalten, Personen verwalten, Caterer verwalten, Lebensmittel verwalten und Orte verwalten. Diese Use Cases sind noch detaillierter betrachtbar, allerdings zur Wahrung der Übersichtlichkeit nicht in diesem Diagramm enthalten. Ebenso wurde der Übersichtlichkeit wegen die Include Pfeile zu den korrespondieren „anzeigen“-Use Cases weggelassen.

Da ein Unterhaltungsmanager nicht die komplette Hochzeit verwalten kann, hat er dementsprechend nicht auf alle Bestandteile des Anwendungsfalles Zugriff. Daher kann nur die Use Cases: Hilfsmittel verwalten, Aktionen verwalten, Orte verwalten und Catere verwalten ausführen.

Ein weiterer Bestandteil ist der seperater Anwendungsfall „Hochzeitsveranstaltung anzeigen“, welcher von allen in diesem Diagramm vorhandenen Akteuren ausführbar ist, da sich alle Benutzer die Daten der Hochzeit anzeigen lassen können. Es gibt jedoch eine Ausnahme: Falls eine Aktion dem Hochzeitspaar verborgen bleiben soll und dies durch den Unterhaltungsmanager so gesetzt ist, kann das Hochzeitspaar diese Aktion nicht sehen. Dieser Use Case enthält folgende weitere Use Cases: Ablaufplan anzeigen sowie Teilnehmer anzeigen.

## Aktion verwalten

Der Use Case „Aktion verwalten“ ist wichtiger Bestandteil der Anwendung, da die Aktionen ein Kernbestandteil der Hochzeit sind. Er besteht im Wesentlichen aus den folgenden Anwendungsfällen: Aktion anlegen, Aktion anzeigen, Aktion ändern, Aktion löschen. Es handelt sich um eine Verfeinerung.

„Aktion anzeigen“ ist ein dieser seperatan Fäller, bei diesem der Akteur, in dem Fall ein beliebiger Benutzer, sich eine Liste der relevanten Aktionen anzeigen lassen kann. Er erbt vom dem Use Case „Liste anzeigen“, was ein generalisierter Use Case zum Anzeigen von Listen ist.

Der Andwendungsfall „Aktion anlegen“ stellt das Anelgen einer Aktion dar, welcher ebenfalls aus kleineren Use Cases besteht, dazu gehören Teilnehmer hinzufügen und Organistor angeben. Diese inkludieren beide weiterhin die Fälle Teilnehmer anzeigen, sowie Liste alle Nutzer anzeigen. Dies ist notwendig damit Nutzer hinzugefügt werden können, beziehungsweise als Organisator angegeben werden können. Der Fall Organisator angeben ist nur von Bewandtnis, insofern der Ersteller nicht der Organistor sein sollte. Weiterhin enthält „Aktion anlegen“ die Use Cases: Ort angeben, Beleg anhängen, Hilfmittel hinzufügen(include auf „Liste aller Hilfmitte anzeigen“, da die Hilfsmittel angezeigt werden müssen, bevor sie hinzugefügt werden können). Ebenfalls würden hier noch der Use Case „… angeben“ für alle Attribute der Entität Aktion vorhanden sein, diese sind aber der Übersichtskeit halber weggelassen.

Der Use Case „Aktion ändern“ dient dazu eine bereits angelegte Aktion zu modifizieren. Dazu kann der Benutzer die einzelnen Attribute der Entität Aktion ändern. Der Anwendingsfall erbt durch einen „extend-Pfeil“ von „Aktion anlegen“, da alle Use Cases dessen auch in diesem Anwendung finden-Zudem sind unteranderem die Anwendungsfälle „Ort ändern“, „Hilfsmittel löschen“, sowie „Teilnehmer löschen“ Bestandteil diese Falles.

Letzlich kann der Benutzer auch mit dem Use Case „Aktion löschen“ angelegte Aktionen wieder entfernen, nachdem sie angezeigt wurden.

## Hilfsmittel verwalten

Der Use Case „Hilfsmittel“ ist ähnlich wie die vorrangehenden Use Cases ein Fall in dem eine Entität verwaltet wird. Das heißt es werden die Prozesse Anzeigen, Erstellen, Ändern und Löschen abgebildet. Akteur ist in diesem Fall ein beliebiger Nutzer, der die entsprechende Berechtigung hat.

Das Diagramm besteht aus vier wesentlichen Bestandteilen: Hilfsmittel löschen, Hilfsmittel anzeigen, Hiflsmittel anlegen, sowie Hilfsmittel ändern. Hilfsmittel anzeigen: Dieser Use Case zeigt eine Liste oder auch nur 1 Hilfsmittel an, dafür gibt es ein extend auf den Use Case „Liste anzeigen“. Der Anwendungsfall Hilfsmittel löschen tritt auf wenn der Nutzer ein Hilfsmittel entfernen möchte, dafür wird ihm die Liste der Hilfsmittel angezeigt, weswegen der entsprechende Use Case ein include hat. Selbes Include gilt ebenfalls für den Anwendungsfall Hilfsmittel ändern, mit welchem der Nutzer die Attribute einer Entität Hilfsmittel verändern kann. Dafür wird ebenfalls eine Anzeige der Hilfsmittel benötigt, wo der korrespondierende Anwendungsfall wieder ins Spiel kommt. Weiterhin besitzt der Use Case includes auf folgende atomaren Anwendungsfälle, welche jeweils das entsprechende Attribut modifizieren: „Titel ändern“, „Beschreibung ändern“, „Kosten ändern“, „Art ändern“, sowie „Beleg ändern“. Letzterer hat ebenfalls Includes auf die Use Cases „Beleg hinzufügen“ und „Beleg entfernen“. Letztlich kann der Akteur ebenfalls ein Hilfsmittel anlegen, wofür der Anwendungsfall „Hilfsmittel anlegen“ existiert. Dieser hat wiederrum includes auf die atomaren Use Cases: „Titel eingeben“, „Beschreibung eingeben“, „Kosten eingeben“, „Art eingeben“ und „Beleg hinzufügen“, welche jeweils das entsprechende Attribut der Entität setzen.

## Caterer verwalten

Das Diagramm Caterer verwalten bildet den Use Case „Caterer verwalten“ ab. Die Verwaltung der Attribute Name sowie Beschreibung wurden zur Wahrung der Übersichtlichkeit vernachlässigt. Akteur ist in diesem Fall ein beliebiger Nutzer, der die entsprechende Berechtigung hat. Es werden allen möglichen Anwendungsfälle dargestellt, die im Rahmen der Verwaltung von der Entität „Caterer“ auftreten können. Dazu gehören maßgeblich die Fälle: „Caterer anzeigen“, „Caterer hinzufügen“, „Caterer ändern“, sowie Caterer löschen“. Der Anwendungsfall Caterer anzeigen hat eine extend auf Liste anzeigen, da er eine List von Catereren anzeigt. Dafür benötigt er Includes auf Use Case „Anzeigen“ der entsprechenden Enitätsmengen, die er als Attribute besitzt. Dazu gehören: „Beleg anzeigen“, „Essen anzeigen“, „Trinken anzeigen“. Der Anwendungsfall „Caterer löschen“ includiert „Caterer anzeigen“, da zum Löschen der entsprechenden Entität sie zunächst dem Nutzer angezeigt werden muss. Ein weiter Use Case ist Catere hinzufügen, in welchem der Akteur eine neue Entität vom Typ „Caterer“ erstellt. Dafür besitzt dieser(Use Case) includes auf „Beleg hinzufügen“, „Essen hinzufügen“ und „Trinken hinzufügen“. Diese Use Cases wiederum besitzen ein include auf die jeweils korrespondierenden „Anzeigen“ Anwendungsfall, da die entsprechenden Entität vor dem hinzufügen angezeigt werden müssen. Der letzte Use Case in diesem Diagramm ist „Caterer ändern“ welcher ein extends auf Caterer hinzufügen hat, da er diesen erweitert. Er erhält zusätzliche includes auf „Beleg entfernen“, „Essen entfernen“ sowie „Trinken entfernen“ , diese hätten jeweils auch ein Include auf den entsprechenden „Anzeigen“ Use Case, diese wurden aber aufgrund der Übersichtlichkeit nicht eingefügt.

## Lebensmittel verwalten

Das Diagramm „Lebensmittel verwalten“ bildet den Use Case „Lebensmittel verwalten“ ab. Akteur ist in diesem Fall ein beliebiger Nutzer, der die entsprechende Berechtigung hat. Es werden allen möglichen Anwendungsfälle dargestellt, die im Rahmen der Verwaltung von der Entität „Lebensmittel“ auftreten können. Dazu gehören maßgeblich die Fälle: „Lebensmittel anzeigen“, „Lebensmittel hinzufügen“, „Lebensmittel ändern“, sowie „Lebensmittel löschen“. Der Use Case „Lebensmittel anzeigen“ hat ein extend auf List anzeigen, da es eine Liste von Lebensmitteln anzeigen kann. Diesen includiert der Anwendungsfall „Lebensmittel entfernen“, da vor dem Löschen einer Entität diese zunächst angezeigt werden muss. Weiterhin existiert der Use Case Lebensmittel hinzufügen, welcher den Prozess abbildet wenn der Nutzer eine neue Entität vom Typ „Lebensmittel“ erstellen möchte. Dafür hat dieser Anwendungsfall includes auf díe atomaren Fälle „Titel hinzufügen“, „Beschreibung hinzufügen“, „Menge angeben“ sowie „Mengenbeschreibung angeben“. Diese setzten jeweils das entsprechende Attribut der Entität. Ähnlich ist der Use Case „Lebensmittel ändern“ aufgebaut. Jedoch erstellt dieser kein neues Objekt, sondern modifiziert ein bestehendes. Dafür besitzt er die entsprechenden Includes auf die Andwendungsfälle „Titel ändern“, „Beschreibung ändern“, „Menge ändern“ sowie „Mengenbeschreibung ändern“. Diese sind ändern jeweils das korrespondiernde Attribut der Entität.