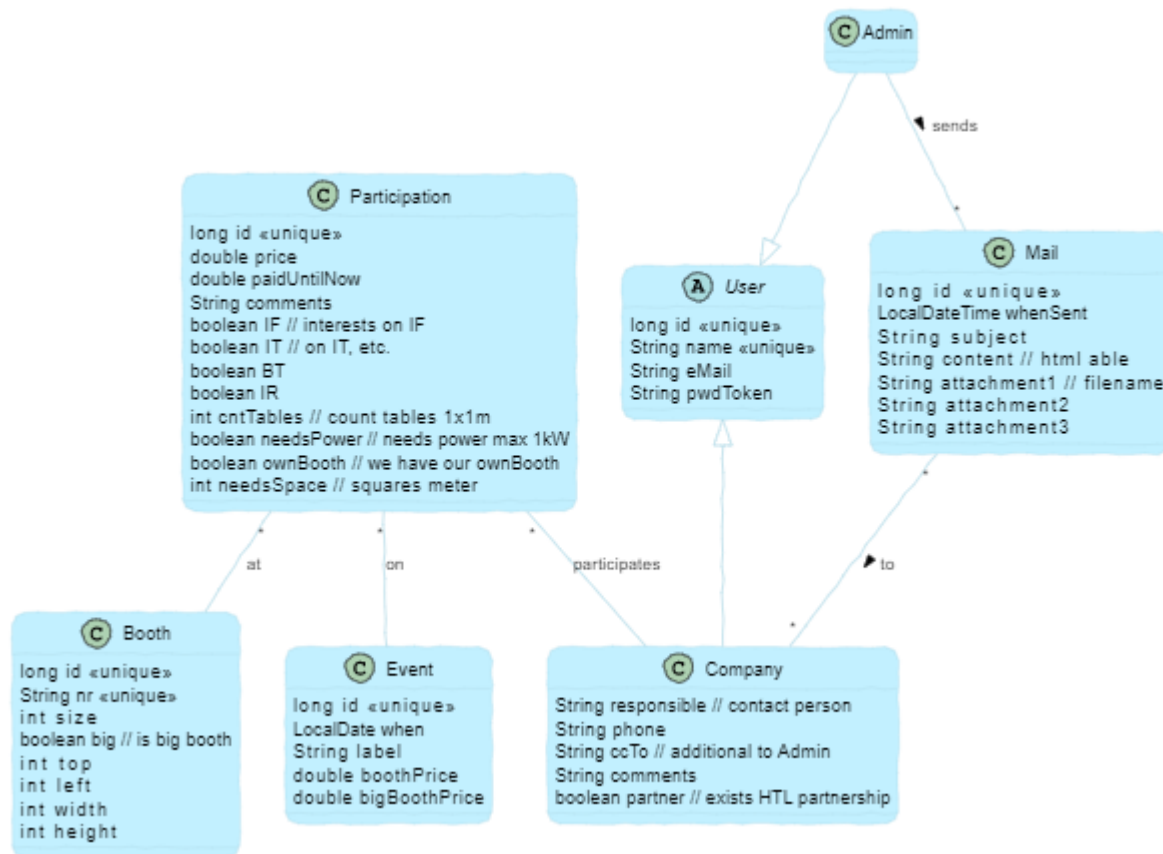


Wirtschaftstag

Wir entwickeln ein System für die Administration von Wirtschaftstagen vor Ort im Haus.

Konzeptuelles Model

Ausgangsbasis ist folgendes konzeptuelle Datenmodell in Form eines UML class diagrams:



Ein Lehrer organisiert als Admin einen Wirtschaftstag. Dazu legt er zunächst einen Event an einem Tag (when) mit einem Namen (label, zB Wirtschaftstag 2025) sowie Preise für normale Stände (boothPrice) und doppelt so große Stände (bigBoothPrice) an. Dann sendet er per eMail (whenSent, subject) eine Einladung (mit content samt optionalen links/attachments) an Firmen. Über ein Formular im link/attachment meldet eine Kontaktperson (responsible) eine Firma zum Event an und legt deren Interessen sowie Equipment an einen Stand fest (IF/./IR, .. needsSpace, etc. – siehe Anmeldeformular unten). Beim nächsten Wirtschaftstag/Event kann der Admin vor dem Aussenden der Einladungs-eMail einige oder alle Firmen auswählen und diesen dem letzten oder geänderten Stand samt Equipment zuordnen. Der Admin kann jederzeit neue Firmen erfassen oder ändern. Dazu gehört die Festlegung der optionalen ccTo-eMail-Adresse. Liegt diese vor, erhält die angegebene eMail-Adresse (EDK, ORG, KAA, ..) als Kontaktperson für diese Firma alle eMails im cc aller vom Admin gesendeten eMails als Kopie (und hoffentlich per replyAll alle Antworten der Firmen zur Information als Kontaktperson).

Die Preise für normale und große Stände legt der Admin pro Wirtschaftstag fest (boothPrice bzw. bigBoothPrice). Beim Anlegen der Participation-Einträge erhalten teilnehmende Firmen ihre jeweiligen Preise (price), je nachdem ob es sich um einen normalen/großen Stand handelt (big). Wirtschaftspartner der HTL (partner) dürfen kostenlos an Wirtschaftstagen teilnehmen.

Unternehmen

Ansprechperson

E-Mail

Telefon

Unser Unternehmen interessiert sich für folgende Fachrichtungen (Mehrfachauswahl ist möglich):

Bautechnik (ca. 180 Schüler)

☐

Innenarchitektur (ca. 80 Schüler)

☐

Informatik und Management (ca. 100 Schüler)

☐

Medien- und Netzwerktechnik (ca. 100 Schüler)

☐

Unser Unternehmen benötigt vor Ort:

Keinen Tisch

☐

1 Tisch (1m x 1m)

☐

2 Tische (je 1m x 1m)

☐

Stromanschluss (maximale Leistung: 1kW)

☐

Wir bringen unseren eigenen Stand mit

☐

Benötigt werden ca. m² Ausstellungsfläche.

Entwickeln Sie dazu entsprechende Controller gemäß Spring Boot Web MVC in Kombination mit einer persistence layer auf Basis Spring Boot JPA. Realisieren Sie für Ihre Controller auch JUnit Tests für die CRUD operations, welche die unten beschriebenen use cases erfordern.

Use-Cases

Jeder der folgenden use cases fasst ein wesentliches Szenario für Akteure des Systems zusammen und beschreibt, was dabei zum Erfolg oder Misserfolg führt. Diese Beschreibungen erfolgen gemäß folgendem tabellarischen Schema:

ID	fortlaufende, eindeutige Nummer
Name	sprechender, eindeutiger Name
Kurzbeschreibung	kurze, aussagekräftige Beschreibung
Actors	alle Akteure, die teilnehmen
Pre-Conditions	alle Bedingungen, die erfüllt sein müssen, bevor der Ablauf gestartet werden kann
Post-Conditions	alle Bedingungen, die gelten, nachdem der Ablauf erfolgreich ausgeführt wurde
Trigger	Ereignis, das diesen use case auslöst und dessen Ablauf startet
Basic / Normal Flow	Beschreibung für den idealen und erfolgreichen Ablauf; ohne Ausnahmen
Alternative Flows	Beschreibungen wichtigster Abläufe bei Ausnahmen oder Fehlersituationen
Test Cases	Beschreibungen wesentlicher Tests

Es gibt folgende wesentliche use cases (in chronologisch relevanter Reihenfolge):

ID	Rolle: <i>ein</i>	User Story: <i>möchte</i>	Grund : <i>damit er</i>
1	Admin	einen neuen Firmentag (FT) anlegen	einen neuen FT administrieren kann add Event
2	Admin	vorhandene Firmen hinzufügen	Firmen voriger FT einladen kann 1. Admin selects Companies comps 2. Foreach c in comps: - add Participation to FT - add c's last booth + equipment - send eMail and add MailLog entry
3	Admin	eine neue Firma einladen (zum FT, dh einen neuen Responsible animieren)	zusätzliche Firmen zu einen FT einladen kann send eMail and add MailLog, die Firma erhält sofort einen Stand samt default Equipment
4	Admin	eMails an angemeldete Firmen senden	alle oder einzelne Firmen eines FT informiert
5	Responsible	eine neue Firma registrieren	seine Firma zu einem FT anmelden kann
6	Admin	Anmeldungen zu einem FT einsehen, ändern und löschen	Anmeldungen eines FT kontrollieren u. eMails als Bestätigungen an Firmen senden kann
7	Admin	eMails einsehen	weiß, wer was per eMail erhalten haben soll, und vice versa, was wer erhalten haben soll
8	Admin	einen FT abrechnen	die Einnahmen mit dem Konto checken kann
9	Admin	offene Abrechnungen sehen	offene Rechnungen als eMail versenden kann
10	Admin	User Daten bearbeiten	Daten aller Arten von User hinzufügen, ändern und löschen
11	Admin	den Standplan eines FT ausdrucken	einen Plan der Stände auflegen kann

Nun folgt eine exemplarische Beschreibung eines use cases. Alle anderen verfügen über das gleiche tabellarische Schema, weshalb diese wegen ihrer selbsterklärenden user story in obiger Übersicht nicht näher ausgeführt werden.

Alle use cases für Administratoren deckt ein mobiler client an. Das Anlegen eines Administrators erfolgt direkt in der Datenbank (durch einen Entwickler mit den Werten wtAdmin, wtadmin@htl-villach.at und wtAdmin#1). Eine web Anwendung deckt die Funktionalitäten aller use cases für Responsibles ab.

Neuen Firmentag anlegen

ID	1
Name	einen neuen Firmentag (FT) anlegen
Kurzbeschreibung	ein Administrator legt einen neuen Firmentag im mobilen client an
Actors	Administrator
Pre-Conditions	es gibt in der Datenbank keinen Firmentag (event) mit dieser Bezeichnung (label) und der Administrator füllt alle Felder aus
Post-Conditions	die Daten wurden über das API in der Datenbank gespeichert und das Datum (date) liegt in der Zukunft
Trigger	ein Administrator drückt im mobilen client Anwendung den Save Button oder schließt das entsprechende GUI
Basic / Normal Flow	ein Administrator meldet sich über den mobilen client an, gibt alle Daten eines neuen Firmentags ein, und er speichert über das API die Daten in der Datenbank
Alternative Flows	es gibt bereits einen Firmentag mit dieser Bezeichnung, oder das API ist nicht erreichbar, oder es fehlen Daten des Firmentags, oder das Datum des Events liegt in der Vergangenheit
Test Cases	Testfall – TC 1 <u>Vorbedingung:</u> ein Administrator hat noch keinen Firmentag mit der gleichen Bezeichnung eingetragen <u>Aktion:</u> ein Administrator will einen Firmentag anlegen, vergisst allerdings das Datum in der Zukunft anzugeben <u>Ergebnis:</u> Fehlermeldung: das Datum muss in der Zukunft liegen

	<p>Testfall – TC 2 <u>Vorbedingung:</u> ein Administrator hat noch keinen Firmentag mit dieser Bezeichnung erfasst <u>Aktion:</u> ein Administrator will einen Firmentag anlegen, vergibt jedoch eine Bezeichnung, die es bereits gibt <u>Ergebnis:</u> Meldung: diesen Firmentag gibt es bereits</p> <p>Testfall – TC 3 <u>Vorbedingung:</u> ein Administrator hat noch keinen Firmentag mit dieser gleichen Bezeichnung angelegt <u>Aktion:</u> ein Administrator will einen Firmentag anlegen, vergisst aber alle Daten zu eingeben <u>Ergebnis:</u> Fehlermeldung: es fehlen Daten zum Anlegen dieses Firmentags, z.B. es fehlt der Preis</p>
--	---

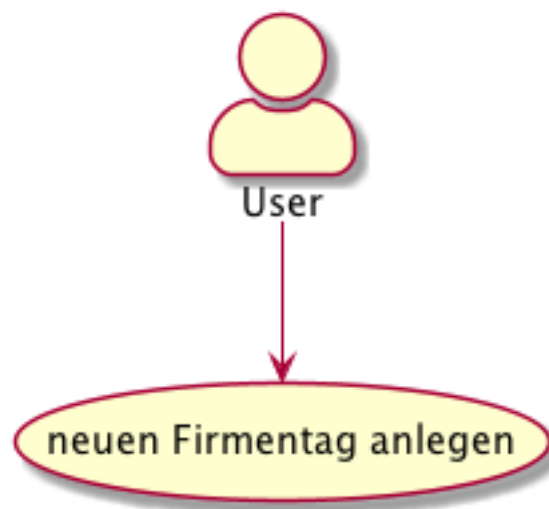


Abbildung 1: Use Case neuen Firmentag anlegen

Neben den MVC-Klassen und deren Tests sind zwei scripts erforderlich. Das script `generate.sh` erstellt Ihre Spring Boot App, wobei alle Tests erfolgreich sein müssen, und ein html file, `zB nn_App.html` (nn als zweistellige Katalognummer), das links zu den nötigen controller Routen in Abhängigkeit obiger use cases beinhaltet. Es deployed außerdem die erforderlichen images in Docker. Mittels `run.sh` starten Sie Ihre App und öffnen im Standard Web Browser Ihres Betriebssystem das generierte oder händisch erstellte `nn_App.html`. In diesem html file sieht der Anwender etwaige erforderliche credentials, zB für die Datenbank, und die dazugehörigen Namen der Docker images, wie zB:

Docker Image	User	Password
SpringBootDB	usr	secret
	admin	topSecret
..		

User Case #	Story	URL: route in localhost:8080/	Hinweise
1	FT anzeigen	api/events	
	FT anlegen	..	
2	Firmen anzeigen	api/companies	

	Firma anlegen	..	
..			

Jeder click auf eine URL zeigt gleich entsprechende Daten oder übermittelt exemplarische Testdaten sofern möglich.

In GUI stehen per default:

- URL vom Webservice
- ausgewählte Firmentag (zB in einer Combo)

UC 1: new Event anlegen und sicherstellen, dass es eindeutig ist und in der Zukunft liegt

UC 2: alle Firmen (in company), die es gibt, einem bestimmten Firmentag (event) zuordnen, dh participations erstellen (default price vom event als price übernehmen)
Vorher alle teilnehmenden Firmen (participations) eines Firmentags (events) anzeigen
Keine Doppelanmeldungen von einer Firma an einem Firmentag

UC 3: neue Firma anlegen (in company) und eine Teilnahme am ausgewählten Firmentag anlegen (participation mit eMail der company für den zuständigen responsible)

UC 4 und 7: Auswahl von Firmen (alle per Knopfdruck und Einzelselektionen), und die Auswahl von Dokumenten als attachments und mit einer bestehenden eMail Applikation am Android device die eMail(s) senden; im (mail) Log eintragen

UC 11: Admins, Responsibles und User bearbeiten (insert, update, delete, list)

UC 10: Anzeigen aller Anmeldungen und per Knopfdruck Auswahl jener Firmen, die noch nicht den Preis vollständig bezahlt haben (paidAlready < price)
optional: Mahnung aussenden per eMail