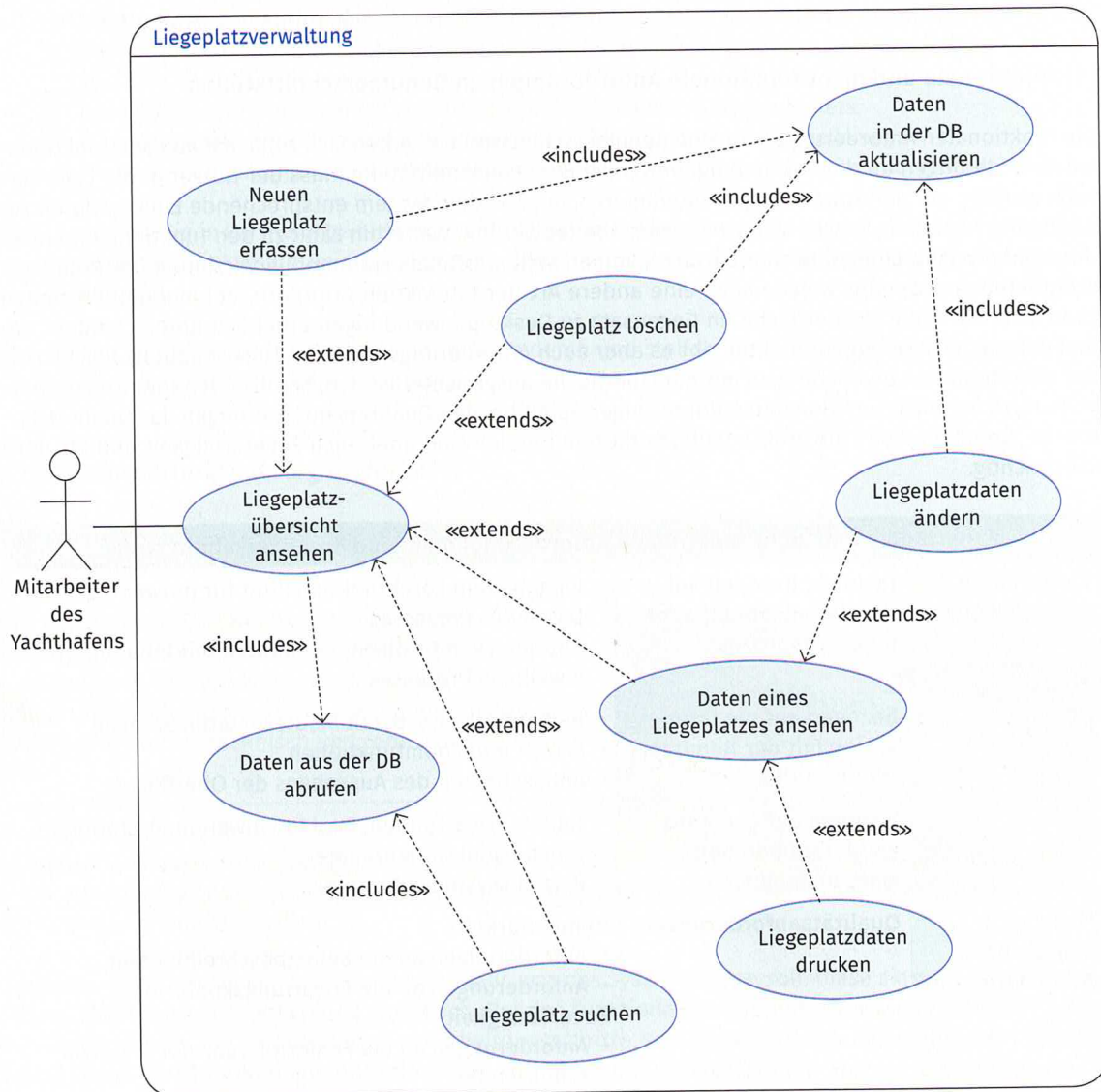


(2) Anforderungskatalog für das Beispielszenario

An dieser Stelle sollen nun exemplarisch die Anforderungen an die Benutzerschnittstelle für die Liegeplatzverwaltung bestimmt werden. Als Grundlage dafür dient das abgebildete Anwendungsfalldiagramm (siehe auch Kapitel 2.1.2 oder Jahrgangsband 2, Lernfeld 8), welches die Grundfunktionalität der Liegeplatzverwaltung beschreibt.



Da ein Anwendungsfalldiagramm auch immer die Interaktion eines oder mehrerer Benutzer mit dem System abbildet, können hieraus unmittelbar Anforderungen an die Benutzerschnittstelle abgeleitet werden. In diesem Fall kann daraus abgeleitet werden, dass die Benutzerschnittstelle u. a. Möglichkeiten zum Hinzufügen und Löschen von Liegeplätzen bereitstellen muss.

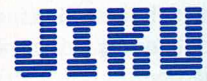
In der folgenden Vorlage hat eine Auszubildende der JIKU IT-Solutions die Anforderungen für die Benutzerschnittstelle für die Liegeplatzverwaltung erfasst.

der Betrachtungen waren dort das Anwendungsfall-, das Klassen- und das Aktivitätsdiagramm. Diese werden in diesem Kapitel noch mal wiederholt und gefestigt. Des Weiteren werden das Zustands- und Sequenzdiagramm neu eingeführt.

Als Grundlage dient wieder ein Fallbeispiel aus dem Projekt „Yachthafen Resort“. In diesem Beispiel sollen auf der Basis einer einfachen Anforderungsspezifikation mithilfe von UML-Diagrammen verschiedene Aspekte einer Softwarekomponente entworfen und modelliert werden.

Zum Modul des Yachthafenmanagements siehe auch S. 112.

Softwareprojekt „Yachthafen Resort“ Liegeplatzverwaltung



Im Rahmen des Projektes soll eine Softwarekomponente für die Vermietung von Liegeplätzen entwickelt werden. Diese gehört zum Modul des Yachthafenmanagements.



Folgende Anforderungen an die Funktionalität werden gestellt:

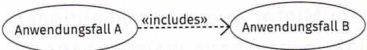
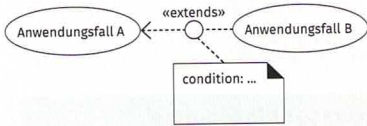
- Kunden sollen einen Liegeplatz mieten und kündigen können.
- Kunden sollen sich im Yachthafen mit ihrer Yacht an- und abmelden können.
- Mitarbeiter des Yachthafens sollen in der Lage sein, einzelne Liegeplätze für Wartungsarbeiten zu sperren.
- Alle Nutzer des Programms sollen sich die aktuelle Liegeplatzbelegung ansehen können.

Um eine hohe Softwarequalität des Programms zu gewährleisten, hat sich das Projektteam dazu entschieden, das Programm objektorientiert zu entwickeln und die moderne Modellierungssprache UML zur Planung zu verwenden.

(1) Anwendungsfalldiagramm

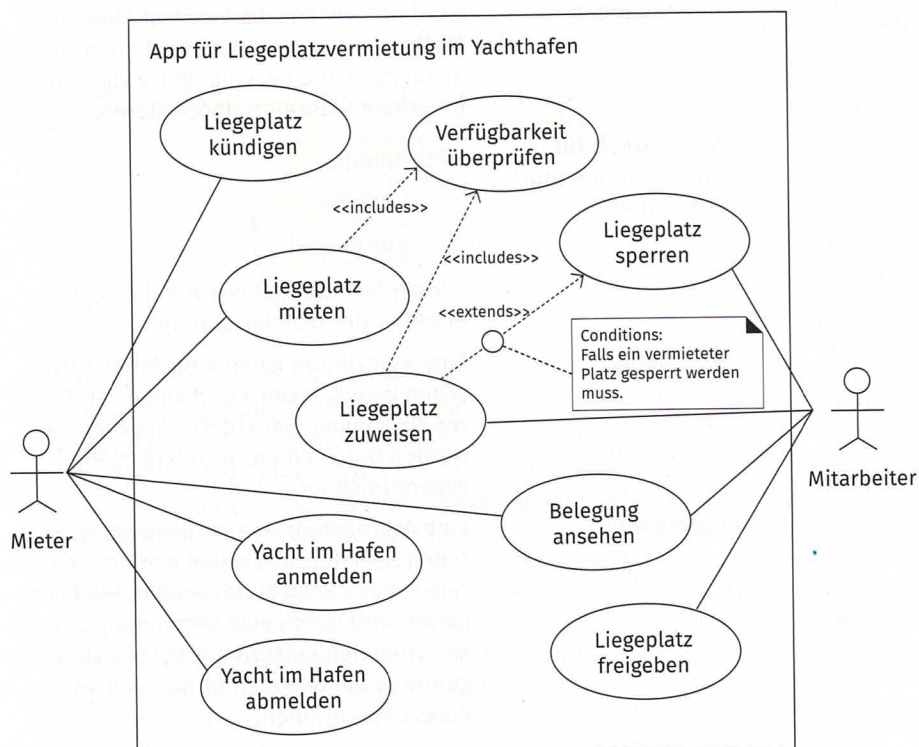
Übersicht über wichtige Notationselemente des Anwendungsfalldiagramms		
Notationselement	Bezeichnung	Beschreibung
	Systemgrenze	Die Systemgrenze umfasst alle Anwendungsfälle, welche das System bereitstellt. Für diese Anwendungsfälle muss später das System die entsprechenden Funktionen bereitstellen.
	Anwender oder Akteur	Hier wird beschrieben, welche externen Benutzer das System verwenden. Dabei ist es auch möglich, dass das System mit anderen externen Systemen interagiert. In diesem Fall wird ein Rechteck statt dem Strichmännchen verwendet.
	Anwendungsfall	Beschreibt eine Aktion oder eine Anzahl von Aktionen, die von dem System für die Benutzer bereitgestellt werden.
	Assoziation	Mit einer Assoziation (Strich) wird eine Beziehung zwischen den Akteuren und den Anwendungsfällen dargestellt.

Übersicht über wichtige Notationselemente des Anwendungsfalldiagramms

	Include-Beziehung	Durch eine Include-Beziehung wird die unbedingte Einbindung der Funktionalität eines Anwendungsfalls in einen anderen Anwendungsfall dargestellt. In dem Beispiel schließt der Anwendungsfall A den Anwendungsfall B ein, d. h., wenn A ausgeführt wird, muss auch B ausgeführt werden.
	Extend-Beziehung	Durch eine Extend-Beziehung wird die bedingte Einbindung der Funktionalität eines Anwendungsfalls in einen anderen Anwendungsfall dargestellt. In dem Beispiel kann der Anwendungsfall A durch den Anwendungsfall B erweitert werden, d. h., wenn A ausgeführt wird, kann unter bestimmten Bedingungen auch B ausgeführt werden.
	Generalisierung/Spezialisierung	Eine Generalisierung/Spezialisierung zeigt die Beziehung zwischen einem allgemeinen und einem spezialisierten Element. Eine Generalisierung/Spezialisierung kann sowohl zwischen Anwendungsfällen als auch zwischen Akteuren dargestellt werden.

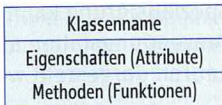
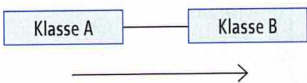
Umsetzung eines Anwendungsfalldiagramms für das Beispiel

Für das Fallbeispiel könnte eine Planung der Anwendungsfälle wie in der Abbildung des Anwendungsfalldiagramms dargestellt aussehen.



Dieses Anwendungsfalldiagramm kann folgendermaßen gelesen werden: Ein Kunde des „Yachthafen Resorts“ soll einen Liegeplatz mieten und kündigen können. Außerdem muss er in der Lage sein, über das System seine Yacht im Hafen an- und abzumelden, wenn er seinen gemieteten Liegeplatz aktiv nutzt. Den Mitarbeitern des Yachthafens soll es wiederum zur Wartungszwecken ermöglicht werden, über das System Liegeplätze zu sperren und wieder freizugeben und notfalls einem Mieter einen Ausweichliegeplatz zuzuweisen. Bevor ein Liegeplatz gemietet oder als Ausweichliegeplatz zugewiesen werden kann, muss immer vom System überprüft werden, ob der entsprechende Liegeplatz auch verfügbar ist. Alle Benutzer des Systems sollen sich zusätzlich über die aktuelle Belegung des Yachthafens jederzeit informieren können. Die Vermietung und Bewirtschaftung von Liegeplätzen wurden hier natürlich vereinfacht dargestellt. Für eine komplexere Aufgabenstellung würde auch das Anwendungsfalldiagramm komplexer ausfallen. Alternativ können auch mehrere Anwendungsfalldiagramme erstellt werden.

(2) Klassendiagramm

Übersicht über wichtige Notationselemente des Klassendiagramms								
Notationselement	Bezeichnung	Beschreibung						
	Klasse	Klassen werden in der UML durch Rechtecke dargestellt, die den Namen der Klasse und/oder die Eigenschaften und Methoden der Klasse enthalten. Klassenname, Attribute und Methoden werden durch eine horizontale Linie getrennt. Der Klassenname steht im Singular und beginnt mit einem Großbuchstaben. Eigenschaften können näher beschrieben werden, z. B. durch ihren Typ. Methoden können ebenfalls durch Parameter usw. zusätzlich beschrieben werden.						
	Alternative Klassendarstellungen	Bei den alternativen Klassendarstellungen werden meistens die Eigenschaften und Methoden weggelassen. Sie dienen vor allem dazu, die Beziehungen zwischen den einzelnen Klassen zu modellieren.						
+, -, #, ~ + schwimmen(): void	Sichtbarkeit für Eigenschaften und Methoden	<table><tr><td>+</td><td>public</td></tr><tr><td>-</td><td>private</td></tr><tr><td>#</td><td>protected</td></tr></table> Mit der Sichtbarkeit werden die Zugriffsmodifizier der OOP beschrieben.	+	public	-	private	#	protected
+	public							
-	private							
#	protected							
	Assoziation gerichtete Assoziation	Eine Assoziation ist eine Beziehung zwischen zwei Klassen. Es ist keine Teil-Ganzes-Beziehung. Gerichtete Assoziationen werden durch einen zusätzlichen Pfeil gekennzeichnet.						
	Aggregation	Eine Aggregation ist eine Beziehung zwischen zwei Klassen, wobei es sich um eine Teil-Ganzes-Beziehung handelt. Die Aggregation wird durch eine Verbindungslinie zwischen den Klassen mit einer nicht ausgefüllten Raute am Ende zur „Ganzes“-Klasse beschrieben.						