

Technology Arts Sciences TH Köln

Medieninformatik Bachelor

Projektdokumentation

Entwicklungsprojekt Interaktiver Systeme WS 17/18

Entwicklung eines Raumbuchungssystems

Webbasiertes Tool mit verteilter Anwendungslogik zur effektiven
Raumnutzung

Abgabetermin: Gummersbach, den 22.12.2017

Prüfungsbewerber:

Bastian Fuchshofer	Niklas Fonseca Luis
Dieringhauser Str. 107	Dieringhauser Str. 107
51645 Gummersbach	51645 Gummersbach

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung der Autoren unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
Listings	VII
Abkürzungsverzeichnis	VIII
1 Einleitung	1
1.1 Nutzungsproblem	1
1.2 Technologie unabhängiger Lösungsvorschlag	2
2 Ziele	2
2.0.1 Einleitung	2
2.0.2 Strategische Ziele	3
2.0.3 Taktische Ziele	3
2.0.4 Operative Ziele	3
2.1 Domänenspezifische Recherche	3
2.1.1 Einleitung	3
2.1.2 Unternehmen	4
2.1.3 Fazit - Unternehmen	4
2.1.4 Lehrinrichtungen	4
2.1.5 Fazit - Lehrinrichtungen	4
2.1.6 Fazit zur Domänen spezifischen Recherche	5
2.2 Marktrecherche	5
2.2.1 Einleitung	5
2.2.2 Locaboo	5
2.2.3 INTIME	6
2.2.4 Online-Raumverwaltung	7
2.2.5 Raumplaner der TH Köln	7
2.2.6 Fazit zur Marktrecherche	8
2.3 Alleinstellungsmerkmale	8
2.4 Risiken	9
2.4.1 Einleitung	9
2.4.2 Projektrisiken	9
3 MCI Inhalte:	9
3.1 Einleitung	9
3.2 Stakeholderanalyse	9
4 Vorgehensmodelle	10

4.1	Einleitung	10
4.2	Auswahl eines Vorgehensmodell	10
4.3	Fazit - Vorgehensmodell	11
5	Benutzermodellierung	11
5.1	Einleitung	11
5.2	User Roles	11
5.3	Focal Roles	13
5.4	Fazit - Benutzermodellierung	14
6	Benutzungsmodellierung	14
6.1	Einleitung	14
6.2	Deskriptive Aufgabenmodellierung	14
6.3	Präskriptive Aufgabenmodellierung	15
6.4	Focal Use Cases	16
6.5	Iterative Evaluation der Aufgaben	17
6.6	Fazit - Benutzungsmodellierung	17
7	Anforderungen	18
7.1	Einleitung	18
7.2	Anforderungen an das System	18
8	Content Modelling	18
8.1	Einleitung	18
8.2	Interface Content Modelling	19
8.3	Context Navigation Map	19
8.4	Fazit - Content Modelling	19
8.5	User Interface Gestaltung	20
9	Gestaltungslösung	21
9.1	Einleitung	21
9.2	Kommunikation des Systems	21
9.2.1	Deskriptives Kommunikationsmodell	21
9.2.2	Präskriptives Kommunikationsmodell	21
9.3	Architektur des Systems	23
9.3.1	Einleitung	23
9.3.2	Deskriptives Architekturmodell	23
9.3.3	Präskriptives Architekturmodell	23
9.4	Fazit - Architektur des Systems	25
9.5	Datenstruktur und relevante Informationen	25
9.5.1	Einleitung	25
9.5.2	Benutzerinformationen	25

Inhaltsverzeichnis

9.5.3	Rauminformationen	26
9.5.4	Filtermöglichkeiten	26
9.5.5	Belegungspläne	26
9.5.6	Sonstige Eigenschaften	27
9.5.7	Fazit - Datenstruktur und relevante Informationen	27
9.6	Standortbestimmung des Benutzers	27
9.6.1	Einleitung	27
9.6.2	Standortbestimmung durch Benutzereingaben	27
9.6.3	Standortbestimmung durch GPS	28
9.6.4	Standortbestimmung durch Bluetooth-Beacons	28
9.6.5	Fazit - Standortbestimmung des Benutzers	29
9.7	Laufwegoptimierung	30
9.7.1	Einleitung	30
9.7.2	Verkettete Liste / Dijkstra-Algorithmus	30
9.7.3	Fazit - Laufwegoptimierung	32
9.8	Flexible Räume (Verteilte Anwendungslogik)	33
9.8.1	Einleitung	33
9.8.2	Aktives aktualisieren von Rauminhalten	33
9.8.3	Passives aktualisieren von Rauminhalten	34
9.8.4	Fazit - Flexible Räume (Verteilte Anwendungslogik)	35
9.9	Verschiedene Eigenschaften der Räume	36
9.9.1	Einleitung	36
9.9.2	Zugang zum Raum	36
9.9.3	Raumtyp 'Stiller Arbeitsraum'	37
9.9.4	Fazit - Verschiedene Eigenschaften der Räume	37
9.10	Personenerkennung im Raum	37
9.10.1	Einleitung	37
9.10.2	Druckplatten im Boden	38
9.10.3	Lichtschranken und Bewegungsmelder	38
9.10.4	Wärmebildkameras	39
9.10.5	Video-/Bildanalyse	39
9.10.6	NFC	40
9.10.7	BLE Beacons	40
9.10.8	Fazit - Personenerkennung im Raum	40
9.11	Fazit	41
Literaturverzeichnis		45
Eidesstattliche Erklärung		46
A	Anhang	i
A.1	Use Cases	i

A.2	Anwendungsfälle (Task Modelling) Version 1	i
A.3	Essential Use Cases Version 1	ii
A.3.1	Administratöraufgaben	ii
A.3.2	Benutzeraufgaben	ii
A.4	Narrative Concrete Use Cases Version 1	iv
A.4.1	Administratöraufgaben	iv
A.4.2	Benutzeraufgaben	vi
A.5	Anwendungsfälle (Task Modelling) Version 2	ix
A.6	Essential Use Cases Version 2	xi
A.6.1	Administratöraufgaben	xi
A.6.2	Benutzeraufgaben	xi
A.7	Narrative Concrete Use Cases Version 2	xv
A.7.1	Administratöraufgaben	xv
A.7.2	Benutzeraufgaben	xvii
A.8	User Roles	xxiv
A.8.1	Focal Roles	xxv
A.9	Content List	xxvi
A.10	Personenerkennung im Raum	xxx
A.11	Druckplatten im Boden	xxx
A.12	Lichtschranken und Bewegungsmelder	xxx
A.13	Zählmechanik	xxxix
A.14	Wärmebildkameras	xxxix
A.15	Video-/Bildanalyse	xxxix
A.16	NFC	xxxix
A.17	BLE Beacons	xxxix
A.18	Ressourcen	xxxix
A.19	Stakeholderanalyse	xxxv

Abbildungsverzeichnis

1	UI Dialog - Reservierung	20
2	Deskriptives Kommunikationsmodell	22
3	Präskriptives Kommunikationsmodell	42
4	Präskriptives Architekturmodell	43
5	3 Schichten Architektur	43
6	Standortbestimmung durch Beacons	43
7	Zugang zum Raum	44

Tabellenverzeichnis

1	Essential Use Cases - Administratöraufgaben	ii
2	Essential Use Cases - Benutzeraufgaben	iii
3	Narrative Concrete Use Cases - Administratöraufgaben	iv
4	Narrative Concrete Use Cases - Benutzeraufgaben	vi
5	Essential Use Cases - Administratöraufgaben	xi
6	Essential Use Cases - Benutzeraufgaben	xii
7	Essential Use Cases - Benutzeraufgaben	xiii
8	Essential Use Cases - Benutzeraufgaben	xiv
9	Narrative Concrete Use Cases - Administratöraufgaben	xv
10	Narrative Concrete Use Cases - Benutzeraufgaben	xvii
11	Der Benutzer möchte die Reservierung für einen von ihm reservierten Raum stornieren - <i>cancelRoomReservation</i>	xxiii
12	Benutzerxxxiii
13	Administratorxxxiv
14	Stakeholderanalysexxxvi
15	Stakeholderanalysexxxvii
16	Stakeholderanalysexxxviii
17	Stakeholderanalysexxxix
18	Stakeholderanalyse	xl
19	Stakeholderanalyse	xli
20	Stakeholderanalyse	xlii
21	Stakeholderanalyse	xliii

Listings

Abkürzungsverzeichnis

Vorwort

Die folgende Projektdokumentation wurde im Rahmen des Moduls Entwicklungsprojekt interaktive Systeme erstellt und befasst sich mit dem Thema der effektiven Raumfindung innerhalb einer Lehreinerichtung und der Planung eines Systems, welches diese Aufgabe unterstützt und für den Benutzer in angemessener Zeit erfüllt. Da das Konzept, welches diesem Projekt vorausgegangen ist, einige Mängel aufgewiesen hat und sich in Teilen nicht für die Umsetzung in einem Projekt durchführen lässt, haben wir uns im Voraus dazu entschlossen dieses zu überarbeiten. Das Nutzungsproblem aus unserem Konzept wurde durch die Berücksichtigung eines Zeitfaktors erweitert. Daraus resultierend wurde auch die Anwendungsdomäne und ein Großteil des Nutzungskontextes überarbeitet. Aus der von uns für das Konzept entworfenen Gestaltungslösung und dem zugehörigen Prototypen wurde der Großteil stark überarbeitet und durch eine neu spezifizierte Gestaltungslösung erweitert. Ziel der Überarbeitung war es aus dem einfachen Buchungssystem des Konzeptentwurfs ein komplexeres Raumbelegungssystem zu entwickeln, welches die automatische Erfassung und Auswertung von Daten ermöglicht.

1 Einleitung

Der Faktor Zeit spielt in vielen Lebenslagen eine entscheidende Rolle! Sei es die Einhaltung von Fristen für Abgaben von Dokumentationen, Präsentationen oder entwickelten Produkten, die genaue Dokumentation von geleisteten Arbeitsstunden oder der Wegstrecke die zurückgelegt werden muss, um von Punkt A nach Punkt B zu gelangen. In vielen Bereichen ist es wichtig Zeit einzusparen, da Zeit bekanntermaßen Geld ist. In nicht erwerbstätigen Bereichen ist der Faktor Zeit dahingehend wichtig, da mit mehr Zeit auch mehr Aktivitäten ausgeführt werden können. Es liegt also nahe, auch in diesen Bereichen die verfügbare Zeit zu optimieren. Ein Punkt an dem dieses Projekt anknüpfen soll, ist das schnelle und unkomplizierte Finden von Räumen die den Erfordernissen und Erwartungen einer Person oder Personengruppe entsprechen.

1.1 Nutzungsproblem

Um die Belegungen von Räumen zu verwalten, benutzen die meisten Unternehmen und Organisationen den klassischen Belegungsplan der sich vor jedem Raum befindet, indem sich Mitarbeiter oder andere Personen über die Belegung von bestimmten Räumen informieren können. Dieser Belegungsplan ist meist eine auf Papier gedruckte Tabelle mit den Tagen und Uhrzeiten wann dieser Raum von wem offiziell belegt ist. Daraus lässt sich dann ableiten, wann ein Raum nicht belegt, und theoretisch von anderen Personen genutzt werden kann. Allerdings sind in diesen Plänen in der Regel nur fest definierte Belegungen, die oft wöchentlich wiederkehren, verzeichnet. Dynamische Nutzungen eines Raumes lassen sich daraus meist nicht ableiten. Die logischste Möglichkeit herauszufinden, ob ein Raum leer ist, ist natürlich diesen einfach zu öffnen und nachzuschauen. Dies setzt allerdings voraus, dass man sich bereits vor solch einem entsprechenden Raum befindet. Um auch Personen die Raumsuche

2 Ziele

zu ermöglichen, die sich nicht zufällig in einem Gang mit möglichen Arbeitsräumen befinden, oder vor einem entsprechenden Raum stehen, wird ein System benötigt, dass über Entfernung überprüft, ob ein Raum zur Verfügung steht oder nicht. Um den Faktor Zeit ins Spiel zu nehmen, wird ein System benötigt, dass in Abhängigkeit der aktuellen Position des Benutzers einen freien Raum angibt, der sich in der Nähe des Benutzers befindet. Muss man sich erst in ein Gebäude, Stockwerk oder Gang begeben um Informationen über den Status eines Raumes einzuholen, wird Zeit in Form von zusätzlichen Laufwegen verschwendet, da die Möglichkeit besteht, dass kein Raum in diesem Gebäude aktuell zur Verfügung steht und man so andere Bereiche aufsuchen muss. Neben einer Zeitersparnis eliminiert man außerdem den Störfaktor der entsteht wenn eine nach einen freien Raum suchende Person einen Raum betritt in dem bereits gearbeitet wird. Die Möglichkeit bei der Raumsuche Zeit zu sparen, die man somit für wichtige Arbeiten verwenden kann, ist also durchaus relevant um effektiver arbeiten zu können. In diesem Projekt soll es um die Einsparung von Zeit in Form von verkürzten Laufwegen bzw. einer verkürzten Suchzeit für einen Raum gehen.

1.2 Technologie unabhängiger Lösungsvorschlag

Um dem Problem der spontanen und unstrukturierten Raumbellegung entgegen zu wirken ist ein System nötig, welches sowohl die verfügbaren Räume, als auch wöchentlich wiederkehrende und spontane Belegungen aufnimmt, um damit einer Person oder Personengruppe auf Raumsuche einen Raum vorzuschlagen. Dieser Raumvorschlag sollte in Abhängigkeit der aktuellen Position des Benutzers erfolgen, um einen kurzen Laufweg gewährleisten zu können. Das System muss den Standort des Benutzers automatisch ermitteln und die Berechnung des Raumvorschlages einbeziehen können. Der Benutzer sollte dabei die Möglichkeit haben sich aus der Ferne über den Status eines Raumes informieren zu können. Bei Bedarf sollte die Möglichkeit bestehen diesen Raum für eine bestimmte Zeitspanne zu buchen, damit gewährleistet wird das kein anderer Benutzer diesen Raum für seine Zwecke verwendet. Dabei sollte darauf geachtet werden, dass der vorgeschlagene Raum auch dem Vorhaben der Person genügt. Eventuelle Bedürfnisse über die Rauminhalte müssen dem System vom Benutzer mitgeteilt werden.

2 Ziele

2.0.1 Einleitung

Anhand des identifizierten Nutzungsproblems und des ermittelten Alleinstellungsmerkmals, lassen sich konkrete Ziele aufstellen die so ein System erfüllen muss um einen qualitativen Lösungsansatz zu erzielen. Die folgenden Ziele sind unterteilt in strategische, taktische und operative Ziele, die zu Teilen während des Entwicklungsprozesses gebildet wurden.

2.0.2 Strategische Ziele

- Das fertige System in unserem Projekt **muss** es dem Benutzer ermöglichen innerhalb von 30 Sekunden einen freien Raum zu erhalten der den Erfordernissen des Benutzers entspricht.
- Dabei **muss** das System alle zugänglichen Räume innerhalb der Lehreinrichtung in Echtzeit adressieren und verwalten können.
- Außerdem **muss** das System Gegenstände die sich in einem Raum befinden und diesen charakterisieren erkennen, adressieren und verwalten können.

2.0.3 Taktische Ziele

- Um eine effektive Raumsuche gewährleisten zu können **muss** der Standort des Benutzers in die Raumauswahl mit einbezogen werden.

2.0.4 Operative Ziele

- Der Benutzer **muss** eine Auswahl an benötigten Gegenständen treffen, die in die Raumsuche mit einbezogen werden.
- Der Standort des Benutzers **wird** automatisch bestimmt.
- Das Betreten/Verlassen eines Gegenstandes aus/in einen Raum **muss** vom System erkannt werden.
- Alle für den Raum relevanten Gegenstände **müssen** markiert werden.
- Vor Betreten eines Raumes **muss** festgestellt werden ob ein Benutzer eine Aufenthaltsberechtigung hat.
- Falls keine Aufenthaltsberechtigung vorliegt **muss** eine solche ggfs. vom System beantragt werden.

2.1 Domänenspezifische Recherche

2.1.1 Einleitung

Das beschriebene Problem der dynamischen Raumsuche kann so gut wie in jeder Organisation auftreten in der es mehrere Räume gibt, die für unterschiedliche Arbeiten genutzt werden können. Vorallem in Organisationen die eine Vielzahl von verschiedensten Räumen besitzen, möglicherweise sogar über mehrere Etagen oder Komplexe verteilt, ist die Gefahr groß, dass eine arbeitsplatzsuchende Person Zeit mit langen Laufwegen verschwendet, da ihr nicht bekannt ist wo sich der nächste frei nutzbare Raum befindet. Das Problem nimmt größere Ausmaße an, je mehr Personen einen freien Raum suchen.

2.1.2 Unternehmen

Dadurch das große Unternehmen in der Regel auch viele Mitarbeiter beschäftigen, und diese Arbeitsmöglichkeiten benötigen, kommt dieser Nutzungsbereich als Möglichkeit in Frage. Mitarbeiter brauchen ggf. Räume für Präsentationen mit speziellem Equipment wie Beamer oder Whiteboards, oder benötigen einen Raum in dem mit mehreren Personen kreativ gearbeitet werden kann. Genauso werden Arbeitsmöglichkeiten benötigt, in dem einzelne Personen in Ruhe ihre Arbeit erledigen können. In Unternehmen herrscht allerdings meistens eine kontrollierte Arbeitsumgebung, wo über den Tag schon geplant wurde, welche Personen oder Personengruppen was für Räume oder Equipment benötigen, da in Projekten in der Regel ein konkreter Ablauf von Tätigkeiten vorliegt. Außerdem werden Arbeitsmöglichkeiten für Mitarbeiter meist dadurch gewährleistet, dass sie einen festen Schreibtisch oder Büro besitzen.

2.1.3 Fazit - Unternehmen

In großen Unternehmen mit vielen Mitarbeitern besteht in der Theorie ein Bedarf an einem System, dass ein besseres Raummanagement und Zeitersparnis liefert, allerdings werden diese Faktoren durch die kontrollierte Arbeitsumgebung von Unternehmen oft selbst gelöst.

2.1.4 Lehreinrichtungen

Lehreinrichtungen mit vielen Schülern, Studenten oder Auszubildenden Personen sind ein guter Kandidat um eine Lösung für ein solches Nutzungsproblem zu finden. Dadurch das Lerner neben festen Veranstaltungen noch beispielsweise Gruppenarbeiten oder Stillarbeit erledigen, ergibt sich eine sehr variable Anzahl an Personen, die im gleichen Zeitraum einen freien Raum benötigen. Die Nutzergruppen beschränken sich also auf die Angestellten und Mitarbeiter der Lehreinrichtung, sowie alle Lerner die interessiert an einer Funktion sind, die es ihnen ermöglicht, schnell und einfach einen freien Raum zu finden, der ihren Ansprüchen genügt. Dabei ist nicht nur das finden von freien Räumen sondern, je nach Kontext, die Information ob ein Raum gerade belegt ist interessant. Eventuell vorhandenes Reinigungs- oder Wartungspersonal kann somit z.B. erfahren, welcher Raum aktuell belegt ist um die darin befindlichen Personen nicht zu stören. Desweiteren können Mitarbeiter die Wartungsarbeiten innerhalb eines Raumes durchführen herausfinden, wann sie für ihre Arbeiten den Raum belegen und nutzen können. Die Laufwege und dadurch das Zeitmanagement von Mitarbeitern kann somit ebenfalls optimiert werden.

2.1.5 Fazit - Lehreinrichtungen

Die Anwendungsdomäne der Lehreinrichtungen mit Fokus auf einer großen Personen und Raumanzahl bietet durch die Vielzahl an Bedürfnissen der einzelnen Benutzer und der benötigten hohen Produktivität der Arbeiten eine gute und sinnvolle Möglichkeit.

2.1.6 Fazit zur Domänen spezifischen Recherche

Im direkten Vergleich von Unternehmen und Lehreinrichtungen bietet die Anwendungsdomäne der Lehreinrichtung mit vielen beteiligten Personen und vielen Räumen das ideale Umfeld für ein System, dass den Benutzern dabei hilft Zeit bei der Suche nach einem freien Raum zu sparen. Für Lehreinrichtungen mit wenigen Räumen und/oder Benutzern, ist dieses System weniger relevant, da eine wirkliche Zeitersparnis in den meisten Fällen nicht gewährleistet werden kann.

2.2 Marktrecherche

2.2.1 Einleitung

Schaut man sich im Internet nach Angeboten zu Raumplanungssystemen um, finden sich viele Angebote die in diese Richtung gehen. Im Zuge einer Marktrecherche wurden verschiedene Anbieter von Raumplanungs-Managementsystemen auf ihre angebotenen Funktionalitäten überprüft, Vor- und Nachteile ermittelt und auf den benötigten Projektkontext hin verglichen.

2.2.2 Locaboo

Locaboo [LOY GMBH, 2017] wirbt mit einem Belegungsplan für Sport-, Bildungs- und Freizeitstätten. Dabei soll die Softwarelösung aus dem Hause LOY GmbH die Verwaltung von verschiedenen Räumen und Bereichen übersichtlich und zeitsparend möglich machen. Genannt werden Funktionen die eine effektivere Auslastung der angebotenen Räume oder Bereiche ermöglichen, sowie die einfache Vermietung von Ressourcen. Das übersichtliche angebotene Dashboard soll eine schnelle und zuverlässige Übersichtsseite der vorhandenen Ressourcen, und der Einstellung von Informationen wie Raumdaten oder Preisquellen ermöglichen.

- Vorteile:
 - funktionales und übersichtliches Dashboard
 - effektives Raum- und Ressourcenmanagement
 - Möglichkeit um für verschiedene Einsatzzwecke Spezialmodule zu buchen
 - Möglichkeit Teilflächen von bspw. Räumen zu buchen.
 - flexible Abrechnungsmethoden
 - Einbindung von Kundenaktivitäten
- Nachteile:
 - Fokus auf Verfügbarkeit und Auslastung von Räumen und Ressourcen

- keine flexible Raumsuche in Abhängigkeit des Standortes des Benutzers

Fazit - Locaboo Diese Softwarelösung ist als ein digitaler Kalender zu sehen der im Prinzip die digitalisierte und standortunabhängige Version des Belegungsplanes vor jedem Raum darstellt. Es gibt zwar die Möglichkeit schnell und einfach Belegungen hinzuzufügen oder zu entfernen, jedoch liegt der Fokus mehr auf der Verwaltung von Räumen aus Veranstaltersicht, und nicht aus der Sicht der einzelnen Raumnutzer.

2.2.3 INTIME

INTIME [COMTEC-NÖCKER GMBH, 2017] von COMTEC bietet ebenfalls die Möglichkeit Räume und Gebäude besser verwalten zu können. Dabei bietet es auch neben zahlreichen Zusatzfunktionen die für Lehreinrichtungen weniger relevant sind, für den einzelnen Benutzer die Möglichkeit schnell einen Raum zu finden der seinen Erwartungen und Erfordernissen entspricht.

- Vorteile:
 - effektives Raum- und Ressourcenmanagement
 - Einzelpersonen oder Gruppen können nach einem Raum suchen
 - Möglichkeit bestimmte Gegenstände zu einem Raum dazu zu buchen
 - Kalenderfunktion
 - Bestandsaufnahmen von vorhandener Technik und Ressourcen
- Nachteile:
 - keine flexible Raumsuche in Abhängigkeit des Standortes des Benutzers möglich

Fazit - INTIME Die Softwarelösung *INTIME* beschränkt sich auf das Verwalten und Managen von Ressourcen. Die Benutzer des Systems sind Personen die auch aktiv einen Raum suchen. Ob sich der Raum allerdings in der Nähe des Benutzers oder am anderen Ende des Gebäudekomplexes befindet ist nicht ersichtlich. Hier wird der Faktor Zeit also nicht vollständig berücksichtigt, der Fokus liegt vielmehr auf der Verfügbarkeitsüberprüfung.

2.2.4 Online-Raumverwaltung

Die Softwarelösung zur Raumverwaltung der Firma *OMOC interactive*) liefert eine cloudbasierte Möglichkeit um es dem Benutzer zu ermöglichen, Raumreservierungen vorzunehmen. Die Software ist von verschiedenen Endgeräten aus bedienbar und bietet alle gebuchten Funktionalitäten komfortabel an. *OMOC interactive* wirbt mit einer leichten Integration auf Websites, Outlook und auf Wunsch auch die Übernahme von bereits bestehenden Daten in das neue System. Die Software bietet eine Verwaltung von Veranstaltungen und Ressourcen ohne Installation und Wartungskosten für den Kunden.

- Vorteile:
 - Verwaltung von komplexen Veranstaltungen
 - Planen und abbuchen von Veranstaltungen
 - von überall bedienbar
 - Cloudbasiert
 - Rechnung/Mahnungen mit einem Klick verschicken
 - Integration der Software in verschiedene Systeme möglich
- Nachteile:
 - keine flexible Raumsuche in Abhängigkeit des Standortes des Benutzers.

Fazit - Online-Raumverwaltung Die Funktionen der Online-Raumverwaltung [[OMOC.INTERACTIVE, 2017](#)] der *OMOC interactive* bietet zwar eine praktische und cloudbasierte Möglichkeit um Räume für Veranstaltungen zu organisieren und verwalten, hier liegt der Fokus aber mehr auf der Vermietung solcher Räume. Der Endbenutzer, mit seinem Interesse im kompletten Prozess der Raumsuche Zeit zu sparen, ist hier nicht adressiert.

2.2.5 Raumplaner der TH Köln

Innerhalb der Anwendungsdomäne der Lehreinrichtungen stellt die Technische Hochschule Köln (Campus Gummersbach) ein Online-Tool [[TH KÖLN, 2009](#)] zur Verfügung, mit dem Einsicht in Raumbelegungen innerhalb des Campus vorgenommen werden können. Es besteht die Möglichkeit Informationen zu bestimmten Räumen einzuholen die unter anderem Belegungen, Zeitrahmen der Belegung und Ausstattung des Raumes beinhalten.

- Vorteile:
 - Übersicht über alle verfügbaren Räume
 - bei vielen Räumen Angaben zum Rauminhalt

2 Ziele

- Start- und Endzeitpunkt der Belegung
- Nachteile:
 - nur wöchentlich wiederkehrende Veranstaltungen
 - lediglich eine Digitalisierung der Aushangpläne vor jedem Raum
 - keine flexible Raumsuche in Abhängigkeit des Standortes des Benutzers

Fazit - Online-Tool der TH Köln Das Online-Tool der TH Köln bietet zwar eine Übersicht über alle vorhandenen Räume innerhalb der Lehreinrichtung und liefert in vielen Fällen auch Angaben über das Equipment der Räume mit, allerdings sind auch hier keine Zeiteinsparungen im kompletten Raumsuchungsprozess möglich. Die dynamische Erfassung von Raumbelegungen ist auch hier nicht gewährleistet.

2.2.6 Fazit zur Marktrecherche

Es gibt viele Softwarelösungen für ein besseres Raummanagement, jedoch ist keine davon darauf ausgelegt, dem einfachen Benutzer einen freien Raum in seiner Nähe anzuzeigen und diesen für ihn zu reservieren / buchen. Fast alle gefundenen Lösungen beziehen sich auf die Vermietung von Räumen oder Veranstaltungsorten die von einem Unternehmen verwaltet werden. Der Anwendungsbereich ist also vorrangig im kommerziellen Bereich. Benutzer sind in den meisten Fällen die Unternehmen selbst, die eine Managementfunktion benötigen. Dabei bieten aber fast alle Softwarelösungen auch die Möglichkeit für einfache Endbenutzer Informationen über einen freien Raum zu bekommen, den sie gerade benötigen. Diese Funktionen beschränken sich aber auf den Verfügbarkeits-Check eines Raumes, unabhängig vom Standort des Raumes zur Person. Den gefundenen Softwarelösungen fehlt somit ein wichtiges Feature um im kompletten Prozess der Raumsuche, beginnend bei dem Verfügbarkeitscheck eines Raumes, über den Laufweg zu diesem Raum, bis zur tatsächlichen Belegung.

2.3 Alleinstellungsmerkmale

Betrachtet man die identifizierten Konkurrenzprodukte, erkennt man das der Verfügbarkeitscheck von bestimmten Räumen bei den meisten Produkten eine kleine, wenn auch nicht unbedeutende, Teilfunktion darstellt. Zeit wird nur dahingehend eingespart, da man sich schnell und einfach über einen freien Raum informieren kann. Die Möglichkeit Zeit zu sparen, indem man kürzere Laufwege und Suchzeiten berechnet, ist in keinem der Produkte namentlich erwähnt. Unser System wird also genau an diesem Punkt anknüpfen um die gefundenen Lücken zu schließen. Konkret bedeutet das, das sich folgendes Alleinstellungsmerkmal für unser System im Vergleich zur gefundenen Konkurrenz ergibt:

- Zeitersparnis durch insgesamt verkürzte Laufwege und Suchzeiten.

2.4 Risiken

2.4.1 Einleitung

Anhand der vorrangegangenen Recherche an Informationen zu einer möglichen Realisierung des Systems wurden erste Risiken identifiziert. Im Verlauf des Projektes wird diese Liste an Risiken bezüglich auf verwendete Technologien und neuen Erkenntnissen stetig aktualisiert. Da noch keine Technologien feststehen die verwendet werden sollen, sind die Risiken aktuell nur auf das Projekt als ganzes bezogen. Sobald Technologien feststehen, und sich darauf neue Risiken ergeben, wird die Liste im Anhang vervollständigt.

2.4.2 Projektrisiken

- Benutzer eines Raumes blockieren den Raum körperlich über die im System hinterlegte Zeitspanne hinaus.
- Das System liefert einen freien Raum nicht in absehbarer Zeit.
- Das Finden eines freien Raumes dauert mit Hilfe der Anwendung länger als ohne.
- Es lässt sich nicht gewährleisten das ein Raum auch tatsächlich frei ist, wenn dem Benutzer dieser ausgegeben wird. (andere Benutzer blockieren den Raum z. B. weil er nicht abgeschlossen ist)

3 MCI Inhalte:

3.1 Einleitung

Im Folgenden beschäftigen wir uns mit den Mensch-Computer Interaktion-Elementen des interaktiven Softwareprojektes.

3.2 Stakeholderanalyse

Um einen ersten Überblick über die möglichen Stakeholder des Systems zu bekommen haben wir im Rahmen einer Stakeholderanalyse potenzielle Benutzer und Interessenten des Systems analysiert.

- Lehrkräfte
- Lerner
- Wissenschaftliche Mitarbeiter
- Leereinrichtungsverwaltung

- Administratoren
- Angestellte
- Lehrereinrichtung

Eine ausführliche Auflistung aller identifizierten Stakeholder mit ihren Eigenschaften und Bezügen zum System befindet sich im (siehe ??).

4 Vorgehensmodelle

4.1 Einleitung

Das anzuwendende Vorgehensmodell in diesem Projekt behandelt im Idealfall die Charakteristiken des Projektfeldes. Dazu zählen z. B. die Projektziele, Aufgaben und organisationale Anforderungen die gegeben sein müssen, um das Projekt effektiv bearbeiten zu können. Im Folgenden soll deduktiv auf ein passendes Vorgehensmodell für dieses Projekt geschlossen werden, welches im Entwicklungsprozess angewendet wird.

4.2 Auswahl eines Vorgehensmodell

Da das zu entwickelnde System dem Benutzer eine Möglichkeit bieten muss ein ganz bestimmtes Ziel zu erreichen, dass innerhalb der Anwendungsdomäne immer das selbe ist, liegt es nahe, dass in der Entwicklung ein besonderer Fokus auf die Modellierung der Aufgaben eines Benutzers gelegt werden muss. Die Aufgaben deren Erfüllung das System gewährleisten muss, beschränken sich auf ein übergeordnetes Ziel, dem Suchen eines Raumes. Die Benutzer und ihre Eigenschaften haben zwar eine gewisse Relevanz, müssen aber nicht wie z. B. im *[User-Centred Design](Buch) von Norman und Draper* im Fokus des gesamten Entwicklungsprozesses liegen. Die Aufführung aller potenziellen Benutzer des Systems schränkt sich, wenn man das Anwendungsumfeld betrachtet, stark ein. Das führt uns zu dem Schluss, dass eine ausführliche Benutzermodellierung in diesem Kontext nicht zwingend notwendig ist. Aus diesem Grund sind auch Vorgehensmodelle die eine starke Einbindung von Benutzern oder Vertretern dieser beinhalten nicht relevant für diesen Projektkontext. Dazu zählen z. B. das *[Scenario Based Usability Engineering](Buch) Vorgehen von Mary Beth Rosson und John M. Carroll*, das mittels einfach zu verstehender Szenarien Benutzern die Möglichkeit bietet effektiv am Entwicklungsprozess teilhaben zu können. Daher sollte der Fokus unseres Projektes eher auf die möglichst effektive Nutzung unseres Systems gelegt werden. Dadurch soll sichergestellt werden, dass der Benutzer oder die Benutzergruppe durch die Verwendung unseres Systems hinsichtlich seines Hauptziels nicht behindert sondern viel mehr unterstützt wird.

4.3 Fazit - Vorgehensmodell

Da in unserem Projekt weder die Einbindung von Benutzern in den Entwicklungsprozess noch die ausführliche Benutzermodellierung in einem benutzerzentriertem Vorgehen ein besonders großen Wert hat, haben wir uns für das *[Usage-Centred Design]*(Buch) von Constantine und Lockwood entschieden. Dieses Vorgehensmodell legt einen besonderen Wert auf die Anwendung des Systems, also die Fokussierung auf den Aufgabenbereich einer Anwendung. Desweiteren ist es durch seine Skalierbarkeit ein ideales Vorgehensmodell für ein Projekt dieser Größe.

5 Benutzermodellierung

5.1 Einleitung

Obwohl im *Usage-Centred Design* von CONSTANTINE UND LOCKWOOD [1999] die Benutzermodellierung keinen sehr hohen Detaillierungsgrad hat, ist dennoch eine Übersicht über Benutzergruppen wichtig um ein effektiv funktionierendes System zu entwickeln. Aus den Benutzergruppen lassen sich benötigte Aufgaben ableiten, die das System abdecken muss. Da wir konkrete Benutzer nur an bestimmten Stellen in den Entwicklungsprozess einbeziehen werden, wenden wir die Benutzermodellierung der User Roles [CONSTANTINE UND LOCKWOOD, 1999, Kapitel 4] an die es uns erlaubt, Benutzergruppen mit ihren Erwartungen an das System zu modellieren. Dabei wird darauf Rücksicht genommen was diese Benutzer-Rollen im Bezug auf ihre Aufgaben innerhalb des Systems darstellen.

5.2 User Roles

Anhand der durchgeführten Stakeholderanalyse lassen sich bereits potenzielle, aktive Benutzer ableiten. Um diese konkret in Rollen unterteilen zu können, haben wir die Erkenntnisse aus der Stakeholderanalyse dafür genutzt in einem Brainstorming Rollennamen zu identifizieren die ihre Aufgabe und ihren Bezug zum System widerspiegeln. Um die Rollen kurz charakteristisch zu beschreiben, haben wir zusätzlich eine Kurzbeschreibung der einzelnen Rollen angefertigt, die die Unterschiede deutlich machen soll. Wir halten das für notwendig, da die einzelnen Rollen sich teilweise nur marginal voneinander unterscheiden. Dabei sind wir auf folgende Ergebnisse gekommen:

- SilentWorkingRoomSearcher
 - sucht einen stillen Arbeitsplatz.
- CasualSingleRoomSearcher
 - sucht einen Arbeitsraum in dem in einer Gruppe gearbeitet werden kann.
- CasualMultiRoomSearcher

- sucht mehrere Arbeitsräume in denen in Gruppen gearbeitet werden kann.
- CasualRoomWithEquipmentSearcher
 - sucht einen Arbeitsraum der spezielles Equipment beinhaltet, das die Benutzer zum arbeiten benötigt.
- RoomWithProfessionalEquipmentSearcher
 - sucht einen Arbeitsraum der spezielles Equipment beinhaltet, das die Benutzer zum arbeiten benötigen.
- SpecificRoomBooker
 - Benutzer möchte einen bestimmten Raum, ohne vorherige Reservierung buchen.
- RoomScheduler
 - möchte den Status eines Raumes abfragen.
 - möchte einen oder mehrere Räume für einen Zeitraum in der Zukunft reservieren/buchen.
- ServerAdministrator
 - möchte den Server des Systems verwalten und aktualisieren.
- DBAdministrator
 - möchte die Datenhaltung des Systems verwalten und aktualisieren.

Während der Ermittlung der Rollennamen sind uns Dopplungen in den Bedürfnissen der Rollen aufgefallen, die wir im nächsten Schritt eliminieren wollen. Dazu haben wir die Rollen neu sortiert und versucht Gruppen zu bilden die ähnliche Interessen besitzen. Dabei ließen sich drei Gruppen mit jeweils unterschiedlichen Spezialisierungen finden.

- FreeRoomSearcher als Gruppe von Benutzern die daran interessiert sind einen Raum zu suchen in dem Arbeiten verrichtet werden können.
 - SilentWorkingRoomSearcher
 - CasualSingleRoomSearcher
 - CasualMultiRoomSearcher
 - CasualRoomWithEquipmentSearcher
 - RoomWithProfessionalEquipmentSearcher
 - SpecificRoomBooker
 - RoomScheduler

- Administrator als Gruppe von Benutzern, die sich um die Verwaltung des Systems kümmert.
 - ServerAdministrator
 - DBAdministrator
- RoomStatusChecker als Gruppe von Benutzern, die Informationen über den aktuellen oder zukünftigen Status eines Raumes benötigen.
 - SpecificRoomBooker
 - RoomScheduler
 - SpecificRoomBlocker

Eine genaue Auflistung der *Role Models* mit ihren Interessen und Bedürfnissen wie im Buch *Software for use* beschrieben, findet sich im ??.

Auch in dieser Auflistung befinden sich noch einige Rollen doppelt, was damit zusammenhängt, dass bestimmte Rollen Eigenschaften besitzen, die die Bedürfnisse von mehreren Rollen widerspiegeln. Sie sind dabei aber nicht als zwei unterschiedliche Rollen zu verstehen, sondern als Spezialisierungen oder Erweiterungen einer bestehenden Rolle. Um diese Abhängigkeiten deutlicher zu machen, haben wir im nächsten Schritt zusätzlich zu den *User Roles* ein Schaubild erstellt. Diese User Role Map [CONSTANTINE UND LOCKWOOD, 1999, Kapitel 4] dient uns dazu, einen Zusammenhang der Rollen untereinander und zum System zu gewähren, damit ein genereller Überblick geschaffen wird der uns hilft im Entwicklungsprozess die Verknüpfungen unterschiedlicher Aspekte nicht aus den Augen zu verlieren. Die *User Role Map* befindet sich zur besseren Lesbarkeit im ??.

5.3 Focal Roles

Die wichtigsten *User Roles* in unserem System sind unserer Meinung nach die Rollen **CasualSingleRoomSearcher**, der nur einen normalen Raum zum Arbeiten benötigt, und der **CasualRoomWithEquipmentSearcher** der einen Raum sucht, der ein bestimmtes Equipment beinhaltet. Da diese Rollen die allgemeinsten, aber wohl auch die häufigsten Rollen in unserem System sein werden, haben wir diese als *Focal Roles* identifiziert. Die Focal Roles [CONSTANTINE UND LOCKWOOD, 1999, Kapitel 4] sind laut [CONSTANTINE UND LOCKWOOD, 1999] die Rollen, die innerhalb eines Systems die typischsten und wichtigsten Benutzergruppen darstellen. Innerhalb unseres Nutzungskontextes sind diese beiden Rollen am häufigsten vertreten und außerdem ausschlaggebend für die Zielerreichung des eigentlichen Nutzungsproblems. Im Entwicklungsprozess haben die Bedürfnisse der *Focal Roles* einen wichtigen Anteil an der Gestaltung des User Interfaces, weswegen sie im Verlauf der Gestaltung immer wieder fokussiert betrachtet werden.

Focal roles are those few user roles judged to be the most common or typical or that are deemed particularly important from some other perspective.

[CONSTANTINE UND LOCKWOOD, 1999, Seite 83]

5.4 Fazit - Benutzermodellierung

Die identifizierten *User Roles* aus unserer Analysearbeit werden im Verlauf des Entwicklungsprojektes dazu genutzt werden die Aufgaben der Benutzer zu spezifizieren und die Anforderungen an das System zu erheben. Außerdem werden die *Focal Roles* dazu genutzt werden den Entwicklungsprozess des User Interface mit einer fokussierten Betrachtungsweise zu unterstützen. [CONSTANTINE UND LOCKWOOD, 1999] schlagen als weitere Ergänzung der Benutzermodellierung das Entwerfen von Structured Role Models [CONSTANTINE UND LOCKWOOD, 1999, Kapitel 4] vor. Wir sind der Meinung das dieser Modellierungsentwurf für dieses Projekt nicht benötigt wird, da eine ausführliche Unterscheidung der Benutzer in den *User Roles* bereits vorgenommen wurde. Die Unterschiede der identifizierten *User Roles* sind minimal, und unterscheiden sich nur in einigen wenigen Punkten, weswegen es unserer Meinung nach nicht notwendig ist diese weiter zu verfeinern. [CONSTANTINE UND LOCKWOOD, 1999] geben weiterhin an, dass der Modellierungsschritt der [CONSTANTINE UND LOCKWOOD, 1999, Kapitel 4] für kleine Projekte oder Projekten mit kurzer Zeitspanne vernachlässigt werden kann [CONSTANTINE UND LOCKWOOD, 1999, Seite 89, 2. Absatz, 3. Zeile], was uns in unserer Meinung zusätzlich bestätigt.

6 Benutzungsmodellierung

6.1 Einleitung

Da im Vorgehen des *Usage-Centred Design* der Fokus auf dem Identifizieren und Verstehen von Aufgaben der Benutzer liegt, haben wir mehrere Modellierungsstufen angewandt um die Aufgaben der Benutzer zu erarbeiten. Dabei sind wir durch immer weiteres und feineres Vorgehen auf die schlussendlich benötigten Aufgaben gestoßen, dessen Erfüllung das System gewährleisten muss.

6.2 Deskriptive Aufgabenmodellierung

Um einen Überblick über die Aufgaben der Benutzer zu erhalten, haben wir zuallererst eine Auflistung der Aufgaben vorgenommen die im aktuellen Nutzungskontext für das Erreichen des Nutzungsziels benötigt werden. Da kein aktives System vorliegt, das diese Aufgaben bearbeitet, sind die meisten der Aufgaben physischer Natur ohne ein technisches System. Die Aufgaben lauten wie folgt:

- Freiern Raum suchen
 - Gebäudeplan studieren
 - Gebäude/Stockwerk/Gang herausfinden in dem sich mögliche Arbeitsräume befinden
 - (optional) Hausmeister/Info/Sekretariat nach freien Räumen fragen
 - (optional) mögliche bereits bekannte freie Räume aus dem Gedächtnis abrufen

- (optional) Raumbellegungspläne, sofern vorhanden, Online prüfen
- zum freien Raum bewegen
- (optional) Raumbellegungsplan vor der Tür auf Belegung überprüfen
- Tür öffnen (sofern möglich) um zu Prüfen ob der Raum frei ist
 - * im Raum nach benötigtem Equipment schauen
- falls der Raum belegt ist, iterieren und ggfs. Gebäude/Stockwerk/Gang wechseln
- falls Equipment nicht vorhanden ist, iterieren und ggfs. Gebäude/Stockwerk/Gang wechseln
- Freien Raum belegen
 - in den Raum begeben und anfangen seine geplante Arbeit zu verrichten
 - ggfs. Raum für wichtige Veranstaltungen wieder freigeben
 - * erneut nach einem Raum suchen

Der Prozess der effektiven Raumsuche gestaltet sich im aktuellen Kontext relativ schwierig, da nur mit Glück das schnelle Finden eines passenden Raumes ermöglicht wird. Im schlimmsten Fall wird bei der Raumsuche viel Zeit verschwendet die anderweitig besser genutzt werden kann. Außerdem gibt es weder eine Garantie das ein Raum frei ist wenn man ihn aufsucht, noch das man überhaupt einen freien Raum in absehbarer Zeit finden kann.

6.3 Präskriptive Aufgabenmodellierung

Um im nächsten Schritt die präskriptiven Aufgaben der verschiedenen Benutzergruppen zu erschließen, haben wir unter Zuhilfenahme der identifizierten *user roles* ein Brainstorming durchgeführt, um einen ersten Überblick über mögliche benötigte Aufgaben bzw. Bedürfnisse der Benutzer zu erhalten. Dabei haben wir uns folgende Fragen gestellt:

- What are users in this role trying to accomplish?
- To fulfill this role, what do users need to be able to do?
- What capabilities are required to support whatever users in this role need to accomplish?
[CONSTANTINE UND LOCKWOOD, 1999, Seite 116,Zeile 7-9]

Auszugsweise im folgenden ein paar exemplarische identifizierte Aufgaben die sich aus dem Brainstorming ergeben haben. Die vollständige Liste der Aufgaben aus dem Brainstorming finden sich im ??.

- Der Benutzer sucht einen ruhigen Raum in dem er Arbeiten kann.

- Der Benutzer sucht einen freien Raum mit bestimmtem Equipment den er nutzen kann.
- Der Administrator möchte einen bestehenden Raum im System mit neuen Informationen anreichern.

Beim Brainstorming haben wir festgestellt, dass einige der Aufgaben starke Ähnlichkeiten besitzen was den Aufbau betrifft. So sind der Hauptteil der Aufgaben dafür zuständig einen Raum zu suchen. Lediglich die Art des Raumes die der Benutzer zum Arbeiten benötigt unterscheidet die Aufgaben. So lassen sich grobe Unterscheidungen bei den Aufgaben treffen die wir im späteren Verlauf genauer beschreiben und darstellen werden.

Da diese Aufgaben teilweise ungenau formuliert, bzw. noch Mehrdeutigkeiten zuließen, haben wir die identifizierten Aufgaben in *Essential Use Cases* formuliert, die die wichtigsten Bedürfnisse der Benutzer und die Reaktion des Systems auf diese Bedürfnisse widerspiegeln ?? . Unserer Meinung nach sind Use Cases ein valides und sinnvolles Mittel um eine Aufgaben-zentrierte Softwarelösung zu entwickeln. Die erarbeiteten *Essential Use Cases* sind laut Definition [CONSTANTINE UND LOCKWOOD, 1999, Kapitel 5] technologieunabhängig und rein auf die Benutzerabsicht und die Reaktion des Systems bezogen, was es uns im späteren Entwicklungsprozess einfacher macht technologiebezogene Gestaltungslösungen zu entwerfen.

Um die erarbeiteten und benötigten Aufgaben evaluieren zu können, haben wir uns dazu entschlossen diese noch in der Form des *Narrative Concrete Use Cases* zu formulieren ?? [CONSTANTINE UND LOCKWOOD, 1999, Kapitel 5]. Diese Use Cases wurden von uns in einer narrativen und verständlichen, in der Anwendungsdomäne befindlichen Sprache verfasst, um sie in einem Evaluationsprozess potenziellen Benutzern vorzustellen. Da narrative Texte für den Benutzer einfacher zu verstehen sind, bietet sich uns damit die Möglichkeit ein paar Benutzer mit den von uns ermittelten Aufgaben zu konfrontieren und ihre Meinung und eventuelle Ergänzungen einzuholen. Da die *Conrecte Use Cases* die Nutzung des Systems ausführlicher beschreiben als die *Essential Use Cases*, wird es für den Benutzer einfacher die Handlungen nachzuvollziehen. Das Feedback der Benutzer wird dazu genutzt die Aufgaben zu überarbeiten und zu verbessern.

6.4 Focal Use Cases

Wie schon in den *Focal Roles* in der Benutzermodellierung, gibt es *Focal Use Cases* die in unserem Fall die wichtigsten bzw. am häufigsten vorkommenden Anwendungsfälle darstellen. Dabei haben wir auf die *Focal Roles* zurückgegriffen und uns überlegt welche Aufgaben diese *Focal Roles* innerhalb des Systems erfüllen müssen. Unserer Meinung nach sind das die Aufgaben **searchingRoomWithSpecificEquipment** und **searchingRoomForSeveralPersons**. Diese Anwendungsfälle haben in unserem System die größte Relevanz da die Funktionalitäten hinter diesen Aufgaben den Grundstein aus Benutzersicht für dieses Projekt legen.

6.5 Iterative Evaluation der Aufgaben

Um die ermittelten Aufgaben zu überprüfen, haben wir ein Gespräch mit 2 Studenten geführt die sich im Umfeld der Anwendungsdomäne befinden und somit potenzielle Benutzer des Systems darstellen. Wir haben ihnen die von uns aufgestellten Aufgaben in der Form der *Narrativ Concrete Use Cases* vorgestellt und sind mit ihnen folgende Fragen für jeden Use Case durchgegangen:

- Können Sie sich vorstellen diese Aufgaben innerhalb des Problemkontextes durchzuführen?
- Welche Vorteile sehen Sie an diesem Anwendungsfall?
- Welche Nachteile sehen Sie an diesem Anwendungsfall?

Nachdem alle Use Cases besprochen wurden und Feedback dokumentiert wurde, haben wir dem Benutzer noch folgende zusätzliche Frage gestellt:

- Können Sie sich weitere Anwendungsfälle vorstellen die bisher nicht erwähnt worden sind?

Wir haben dabei in Kooperation mit den Befragten Benutzern festgestellt das es gewisse Risiken für die Umsetzung des Systems gibt. Zum Beispiel ist nicht gewährleistet das ein Raum auch tatsächlich frei ist wenn der Benutzer ihn vom System vorgeschlagen bekommt und anschließend aufgesucht hat. Es wird also eine Art Reservierungsfunktion benötigt die einen Raum für andere Benutzer temporär als nicht verfügbar kennzeichnet, bis der tatsächliche Benutzer diesen Raum körperlich erreicht hat. Zusätzlich wird die Möglichkeit der vorzeitigen Stornierung eines Raumes benötigt, falls sich der Benutzer spontan entschließt den Raum nicht mehr zu benutzen. Eine Verlängerung der Buchung eines Raumes ist ebenfalls von Vorteil, da es durchaus realistisch ist das ein Benutzer oder eine Benutzergruppe länger benötigt, als die eigentliche Buchung im System hinterlegt ist. Die identifizierten Risiken bzw. Änderungen sind zwar wichtig für einen funktionierenden Ablauf des Systems, allerdings haben wir sie nicht als zusätzliche *Focal Use Cases* definiert. Um diese Abhängigkeiten trotzdem deutlich zu machen haben wir, wie bei den *User Roles* auch, ein Schaubild erstellt. Diese [CONSTANTINE UND LOCKWOOD, 1999, Kapitel 5] befindet sich zur besseren Lesbarkeit im ??.

Wir haben dieses Feedback dafür genutzt die Anwendungsfälle zu überarbeiten und neue Anwendungsfälle in Form der bereits bekannten *Essential Use Cases* zu erstellen. Die Aktualisierungen der Anwendungsfälle haben wir im ?? dokumentiert. Zusätzlich befindet sich im ?? die Dokumentation der wesentlichen Ergebnisse der Befragung.

6.6 Fazit - Benutzungsmodellierung

Das Ergebnis der Aufgabenmodellierung für das System haben wir in Zusammenarbeit mit einigen potenziellen Benutzern ermittelt. Die befragten Personen haben uns Feedback über die Relevanz der ermittelten Aufgaben gegeben, welche wir für den weiteren Entwicklungsprozess benötigen. Die identifizierten und in Use Cases formulierten Aufgaben stellen für den Benutzer die Möglichkeiten da, mit

dem System zu interagieren. Dabei werden vermutlich im fertigen System auch mehrere Aufgaben kombinierbar sein. Die genaue Implementation der Aufgaben innerhalb der Gestaltungslösung wird in einem späteren Abschnitt behandelt. Anhand der Aufgaben lassen sich Risiken ableiten die das Projekt gefährden können. Diese Risiken haben wir im [Anhang](Risiken bei der Aufgabenbestimmung) dokumentiert und zusätzlich angegeben wie wir mit ihnen umgehen.

7 Anforderungen

7.1 Einleitung

Anhand der *Role Models* aus der Benutzermodellierung und den Use Cases aus der Benutzungsmodellierung lassen sich die Erfordernisse und Erwartungen der Benutzer gegenüber dem System beschreiben. Diese identifizierten Erfordernisse werden von uns im nächsten Schritt dazu verwendet Anforderungen für das System zu formulieren.

7.2 Anforderungen an das System

Im ersten Konzept des Projektes wurden bereits erste Anforderungen formuliert. Diese sind unterteilt in Funktionale und Non-Funktionale Anforderungen. Da im Verlauf der Konzeption und Entwicklung einiges abgeändert werden musste, wurden auch diese Anforderungen noch einmal überarbeitet. Zusätzlich haben wir sie um einige wichtige, auf das neu zu entwickelnde System bezogenen Punkten erweitert. Die vollständige Liste der identifizierten Anforderungen des Systems und des Projektes befindet sich im ??.

8 Content Modelling

8.1 Einleitung

Mit Hilfe der Benutzer- und Benutzungsmodellierung kann durch verschiedene Methoden der "Interface Contents and Navigation"Thematik ein übersichtlicher abstrakter Prototyp erstellt werden. Ergebnis dieser Vorgehensweise ist eine "Content List"und eine "Content Navigation Map"mit deren Hilfe die wichtigen Merkmale und Funktionen sowie die Übergänge zwischen den einzelnen Aktionen und Dialogen innerhalb der Benutzerschnittstelle dargestellt werden können. Wichtig hierbei ist, das es sich nur um einen abstrakten Prototypen handelt, also das die Ergebnisse so wenig Designmerkmale und Lösungsvorschläge beinhalten wie möglich. Weiterhin ist zu beachten, das genannte Eigenschaften wie ein Drop-down Menü"nur ein Beispiel sind und für die fertige Gestaltungslösung noch abgeändert werden können.

8.2 Interface Content Modelling

Sinn der Content List ist es eine übersichtliche Auflistung der verschiedenen interaction spaces mit ihren zugehörigen Inhalten zu erstellen. Ein interaction space wird dabei in die verschiedenen Dialoge aufgeteilt. Im Falle dieses Projektes werden die Interaction spaces *userSpace*, *instituteSpace* und *adminSpace* mit ihren Dialogen untergliedert ([siehe Content List]()). Der *userSpace* besteht zum Beispiel aus den folgenden Dialogen:

- Start Dialogue
- Filter Dialogue
- Reservation Dialogue
- Multi Reservation Dialogue
- Booking Dialogue
- Multi Booking Dialogue

Der erste Teil der Dialogbezeichnung wurde so gewählt das durch diesen verdeutlicht wird, welche Aktion diesem Dialog vorausgegangen ist. Der Dialog **Reservation Dialogue** beschreibt zum Beispiel den Kontext den der Benutzer sieht, nachdem er erfolgreich einen Raum reserviert hat. Die Dialoge werden weiterhin durch benötigte Inhalte erweitert und untergliedert. Dabei wurde in diesem Projekt darauf geachtet, dass so wenig endgültige Designeigenschaften der Elemente vorgegeben werden.

8.3 Context Navigation Map

Als Erweiterung zur Content List wurde eine ???. Diese verdeutlicht die Übergänge zwischen den einzelnen Dialogen. In diese Context Navigation Map wurden Vorschläge für verschiedene Interaktionsmöglichkeiten eingetragen. So kann die Beschriftung [**Buchen**] an einem Übergang im späteren Projekt durch einen Button mit der Beschriftung "Buchen" umgesetzt werden. Bei diesen Beschriftungen handelt es sich allerdings nur um einen Vorschlag und keinen fertigen Entwurf, d.h. die Interaktion bzw. der Übergang kann nach dem Interfacedesign auch durch eine andere Aktion aktiviert werden.

8.4 Fazit - Content Modelling

Die aus dem Content Modelling entstandenen Artefakte können für das Interfacedesign verwendet werden, da durch diese die wichtigen Informationen und Übergänge zwischen verschiedenen Dialogen aufgezeigt werden. Die Artefakte wurden allerdings so wenig vom Endgültigen Design vorgegeben wurde.

Abbildung 1: UI Dialog - Reservierung

8.5 User Interface Gestaltung

Für das Design des User Interfaces, kurz UI, haben wir uns in diesem Projekt an den fünf Regeln und sechs Prinzipien der Nutzbarkeit [CONSTANTINE UND LOCKWOOD, 1999] orientiert. Daraus ist als erste UI-Lösung ein Interface entstanden welches verschiedene Aufgaben und Funktionalitäten gebündelt in Oberkategorien zur Verfügung stellt. Damit ein Benutzer z. B. einen Raum buchen kann muss er den folgenden Interaktionspfad durchlaufen.

Hauptmenü > Raum suchen > Raum / Räume suchen > Filter > Reservierung > Buchung

Um den Benutzer innerhalb dieses Interaktionspfades zu leiten haben wir uns für den Stil eines Natural Language Interfaces entschieden, welcher einen Benutzer mit Hilfe von ganzen Sätzen durch eine Aufgabe führt. So entsteht z. B. für einen Benutzer der erfolgreich einen Raum für sich reserviert hat das folgende Interface:

Dieser Aufbau ermöglicht es erfahrenen Benutzer die wichtigen Informationen auf einen Blick zu erfassen, hilft aber unerfahrenen Benutzern dennoch bei der Durchführung ihrer Aufgabe.

Abschließend haben wir das entstandene UI selbstkritisch anhand der Regeln und Prinzipien bewertet und die Mängel aufgeschrieben. Eines dieser Mängel war die nicht gegebene Selbstbeschreibungsfähigkeit also das ein Benutzer ohne Erfahrungen mit unserem System nicht weiß was er zu drücken hat um z.B. mehrere Räume zu reservieren und anschließend zu buchen. Das UI wurde hinsichtlich der von uns festgestellten Mängel verbessert. In dem aus dieser Iteration entstandenen UI findet der Benutzer alle Verfügbaren Funktionen unseres Systems direkt auf der Startseite, sodass er die Lösung zu seiner Aufgabe sofort finden kann. Dadurch hat sich der Interaktionspfad für die Buchung eines Raumes folgendermaßen verändert.

Hauptmenü > Einzelnen Raum suchen > Filter > Reservierung > Buchung

Außerdem wurden Aufgaben wie Einzelne und Mehrere Räume suchen von einander getrennt was die Suche nach diesen Funktionen vereinfacht.

Das aus der Überarbeitung der Mängel entstandene UI wurde dann von Benutzern ebenfalls im Hinblick auf die fünf Regeln und sechs Prinzipien ([s.o.]()) evaluiert. Die aus dieser Evaluation entstandenen Funktionswünsche und Anmerkungen haben wir dann als Basis genommen um das existierende UI zu erweitern. Das daraus entstandene UI verbessert die Übersicht einzelner Interaktionsschritte in Bezug auf das Reservieren und Buchen von Mehreren Räumen. So besitzt ein Benutzer der Mehrere Räume sucht nun die Möglichkeit die Filter für jeden Raum einzustellen und kann anschließend die Reservierung der Räume einzeln und getrennt von einander durchführen. Ebenso kann ein Benutzer welcher bereits Mehrere Räume gebucht hat diese unabhängig voneinander Stornieren und somit wieder im System freigeben.

9 Gestaltungslösung

9.1 Einleitung

Da die Grundbausteine für die Entwicklung einer optimalen Gestaltungslösung durch die Benutzer und Aufgabenmodellierung sowie die Aufführung der Content Elemente nun gegeben sind, geht es jetzt darum eine konkrete Lösung für das Nutzungsproblem zu finden. In einem Brainstorming über den groben Aufbau des Systems haben wir mehrere Probleme identifiziert, auf die in diesem Teil der Dokumentation eingegangen werden soll. Darunter zählen z.B. eine Standortbestimmung der Benutzer innerhalb des Gebäudes um einen konkreten Raumvorschlag liefern zu können, oder auch die Verifizierung der Anwesenheit von Gegenständen oder Personen innerhalb von Räumen. Da es je nach Umfeld sein kann, dass es Räume gibt die nicht von jeder Personengruppe genutzt werden kann oder darf, muss die Zugangsberechtigung für Räume geklärt werden, so das gewährleistet wird das nicht jeder Zugang zu allen Räumen hat. Außerdem muss geprüft werden wie viele Personen sich in einem Raum gerade aufhalten um Aufschluss über die Auslastung der Räume zu bekommen. Diese Informationen sind außerdem wichtig um festzustellen ob ein Raum gerade belegt ist oder nicht. In diesem Teil der Dokumentation befassen wir uns mit der Erarbeitung und kritischen Diskussion von möglichen Technologien für einzelne Unterprobleme, die Zusammenstellung der Komponenten des Systems und ihre Abhängigkeiten und Kommunikationsweg untereinander.

9.2 Kommunikation des Systems

9.2.1 Deskriptives Kommunikationsmodell

Da im aktuellen Kontext kein technisches System existiert haben wir anhand der ?? ein Modell entwickelt, dass die Kommunikation im aktuellen Kontext darstellen soll.

9.2.2 Präskriptives Kommunikationsmodell

Im zu entwickelnden System haben wir uns dazu entschlossen eine REST-Konforme Architektur zu verwenden. Dabei basiert unsere Kommunikation auf dem Client-Server Paradigma. Wie in ?? dargestellt hat der Client den aktiven Part und stellt Anfragen an den Server. Dabei haben wir auf die Unterscheidungen der einzelnen Aufgaben geachtet. Ein Kommunikationsaufbau vom Server zum Client halten wir in diesem Kontext für nicht sinnvoll, da zu viele Endgeräte der Benutzer existieren um sie gezielt durch den Server ansprechen zu können. In unserem Kontext ist dies aber auch nicht nötig da der Server lediglich als Verarbeitungseinheit angesehen wird und keine Anfragen an den Client senden muss. Um REST-Konformität zu gewährleisten wird serverseitig kein Zustand gespeichert. Jede Anfrage des Benutzers liefert alle benötigten Informationen für den Server mit. Dazu gehören auch die Verifikation des Benutzers durch die E-Mail Adresse. Fragt der Client einen freien Raum an, werden die E-Mail Adresse und die eventuell benötigten Gegenstände des Benutzers im Body des

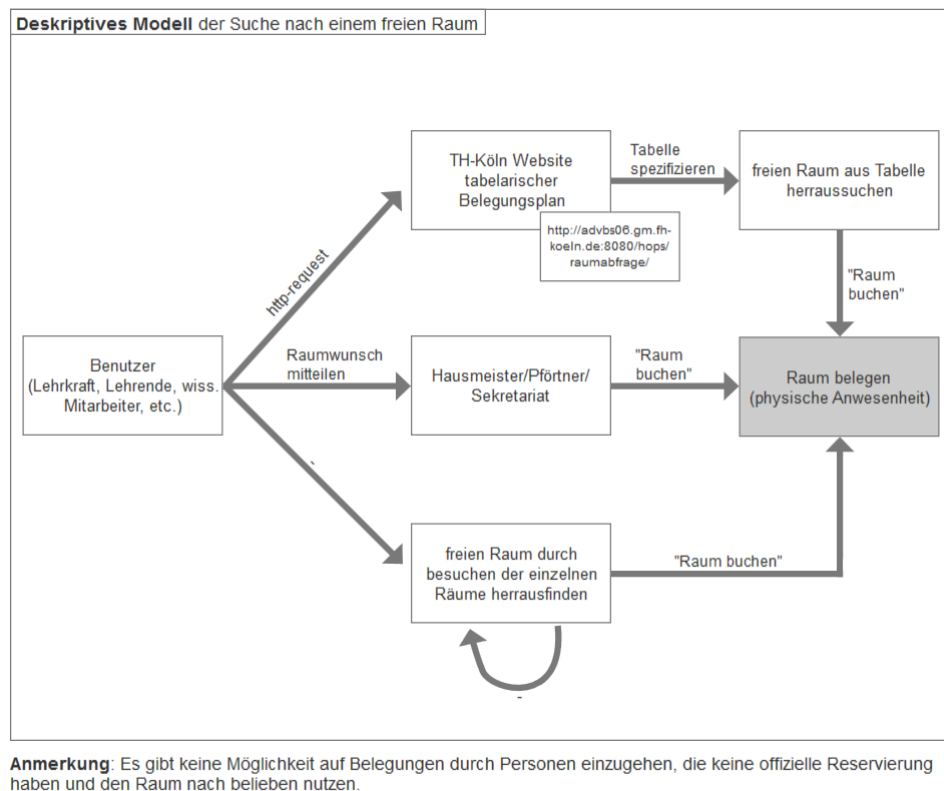


Abbildung 2: Deskriptives Kommunikationsmodell

Requests an den Server gesendet. Nach einer Verarbeitung dieser Informationen sendet der Server einen gefundenen Raum als Response an den Client zurück. Ist der gefundene Raum an den Benutzer ausgegeben, speichert der Server diese Information in der Datenbank. Bei Änderungen bezüglich den Status dieses Raumes, werden die Informationen in der Datenbank aktualisiert. Hat der Benutzer eine Raumnummer vom Server empfangen, werden diese Informationen im Gerätespeicher des Endgerätes zwischengespeichert. Durch das Caching dieser Informationen sind sie noch vorhanden, sollte der Benutzer die Anwendung aus versehen schließen. Die Anwendung hat dann die Möglichkeit diese Informationen wiederherzustellen. Um diese auf Aktualität zu prüfen sollte ein Zeitstempel gespeichert werden anhand dem verglichen werden kann ob die Anfrage noch aktuell ist. Damit sparen wir uns einen zusätzlichen Request zum Server um die Aktualität zu prüfen. Die verbleibende Gültigkeitsdauer einer Information muss dabei sowohl auf Serverseite als auch beim Client gleich sein. Für die Adressierung der Ressourcen benutzen wir einfache Bezeichner die sich je nach Inhalt des Bodys oder eines zusätzlichen Query String unterscheiden lassen. Für den Hauptteil der Benutzer sind die Ressourcennamen jedoch unsichtbar, da über eine Applikation keine sichtbare URI angesteuert wird. Lediglich der Administrator mit einem User Interface über einen Webbrowser hat in unserem System die Möglichkeit Einsicht in die Ressourcenbenennung zu nehmen. Anhand der identifizierten Aufgaben aus der ?? haben wir eine Ressourcenadressierung vorgenommen. Die identifizierten Ressourcen sind nicht endgültig und können sich im Entwicklungsprozess der Gestaltungslösung oder der Implementation des vertikalen Prototyps noch ändern. Die Auflistung der Ressourcen und ihre Adressierungen befinden sich im ??.

9.3 Architektur des Systems

9.3.1 Einleitung

Um die Kommunikation und die Architektur der Komponenten des Systems deutlich darzustellen, haben wir Schaubilder erstellt die es ermöglichen einen Eindruck über unseren Aufbau des Systems zu erhalten.

9.3.2 Deskriptives Architekturmodell

Da es aktuell kein technisches System gibt auf dem wir uns in unserem Projekt beziehen können, lässt sich auch kein deskriptives Architekturmodell ermitteln.

9.3.3 Präskriptives Architekturmodell

Um ein präskriptives Architekturmodell aufzustellen, haben wir zuerst Komponenten identifiziert die in unserem System relevant sind.

Webserver Als Server dient uns ein über [Node.js](https://nodejs.org/en/) aufgesetzter Webserver mit dem [Express Framework](http://expressjs.com/de/) der mittels HTTP-Requests mit den anderen Systemkomponenten kommunizieren kann. Da Node.js von Haus aus asynchrone Kommunikation anbietet, ist es uns möglich mehrere Anfragen gleichzeitig auszuwerten. Der Webserver dient als zentrale Verarbeitungseinheit für das System und behandelt alle eingehenden Anfragen der Benutzer. Die Kommunikation funktioniert sowohl von Richtung Client als auch in Richtung Datenbankserver über einfache HTTP-Requests die im besten Fall über ein geschlossenes Netzwerk innerhalb der Lehrereinrichtung arbeiten. Damit besteht die Voraussetzung das der Benutzer sich im selben Netzwerk wie das System befinden muss, bspw. über eine aktive WLAN-Verbindung. Damit würde eine zusätzliche Authentifizierung der Benutzer erfolgen, da für gewöhnlich nur Personen Zugang zu einer WLAN-Verbindung haben, die dort auch Mitglied oder angestellt sind. Da aber wahrscheinlich nicht alle Lehrereinrichtungen eine WLAN-Verbindung stellen können, sollte der Zugang zum System theoretisch auch über eine Mobilfunkverbindung mit dem Internet möglich sein. Als Grundvoraussetzung um das System nutzen zu können, ist aber trotzdem eine aktive Internetverbindung an die sich der Webserver anknüpfen kann. In unserem Projekt gehen wir davon aus das die Lehrereinrichtung sowohl einen verfügbaren Internetanschluss als auch eine WLAN Verfügbarkeit überall im Gebäude gewährleisten kann.

Datenbankserver Der Datenbankserver des Systems dient als Verbindungselement zwischen Webserver und Datenbank. Dabei liest der Datenbankserver die benötigten Informationen des Webserver aus der Datenbank aus und leitet sie weiter. Außerdem ist er für das abspeichern von neuen oder aktualisierten Informationen zuständig die ihm der Webserver übermittelt. Wir haben festgestellt das logische Unterscheidungen bei den Datensätzen getroffen werden können. Wir haben eine logische Trennung der regulären und der dynamischen Raumbellegung getroffen um eine eventuell bereits bestehende Datenbank mit regulären Veranstaltungsdaten in das System eingliedern zu können. Es bietet sich an diese auch in logisch getrennten Datenbanken einzuordnen um eine Wartung dieser zu vereinfachen. Auf der einen Datenbank sind nur Informationen vorhanden die bereits im Systemumfeld vorliegen, wie z. B. der reguläre Veranstaltungsplan der regelmäßig wiederkehrenden Veranstaltungen speichert. Die andere Datenbank enthält die von uns dynamisch erzeugten Informationen über die aktuelle Belegung der Räume durch die Benutzer des Systems. Durch die Trennung dieser Datensätze kann je nach Bedarf der Inhalt der Datenbanken verändert werden, ohne den Ablauf der anderen Datenbank zu beeinflussen. Als Datenbank haben wir uns für [Redis](<https://redis.io/>) entschieden, die eine einfache Schlüssel-Werte-Datenstruktur besitzt. Da keine komplexen Datenstrukturen benötigt werden ist diese Datenbank ideal für uns geeignet. Ein Vorteil von Redis sind die schnellen Zugriffszeiten die bei ca. 100.000 Schreibvorgängen/s und ca. 80.000 Lesevorgängen/s liegen können was die Bearbeitung von mehreren Anfragen vereinfacht.

Client Die 3. Komponente auf Clientseite ist eine Anwendung auf dem Endgerät des Benutzers. In unserem Projekt haben wir uns auf eine Applikation, basierend auf dem Android Betriebssystem fokussiert. Eine Umsetzung auf einem anderen Betriebssystem, wie z.B. iOS ist aber auch denkbar, wird in diesem Projekt aber nicht behandelt. Laut einer Studie der [Bitkom research](<https://www.bitkom.org/Presse/Anh-an-PIs/2017/02-Februar/Bitkom-Pressekonferenz-Smartphone-Markt-Konjunktur-und-Trends-22-02-2017-Praesentation.pdf>) benutzen 78% der befragten Personen ab 14 Jahren im Januar 2017 ein Smartphone. Mit einer Nutzerzahl von 54 Millionen Menschen in Deutschland (Tendenz steigend) bietet es sich an das Smartphone oder Tablet in den Prozess einzubeziehen. Zusätzlich ergibt sich der Vorteil das eine Standortabhängige Nutzung des Systems ermöglicht wird. So müssen nicht auf z.B. Terminals oder Anzeigetafeln an festen Positionen im Gebäude, Informationen über Räume bezogen werden. Diese Anwendung stellt die Kommunikations- und Interaktionsschnittstelle mit den Funktionen des Systems da. Der einfache Benutzer des Systems kann durch die Installation dieser Anwendung auf seinem Android basierten Endgerät wie z. B. Smartphone oder Tablet die Funktionalitäten des Systems nutzen. Die Anwendung kommuniziert dabei über HTTP-Request mit dem Server des Systems und sendet Informationen die zur Berechnung einer Raumauswahl benötigt werden. Dabei sollte der Anwendung bekannt sein, unter welcher Adresse sich der Webserver befindet. Um das dynamisch und auf verschiedene Lehrinrichtungen einfach anpassen zu können, sollte unter anderem diese Adresse bei der ersten Konfiguration in der Anwendung gespeichert werden. Da sich der Benutzer sowieso durch z.B. Eingabe einer E-Mail-Adresse identifizieren muss, kann durch ein zusätzliches Eingabefeld auch die Webadresse des Servers vom Benutzer erfragt werden.

9.4 Fazit - Architektur des Systems

Die von uns erarbeiteten Komponenten des Systems bilden einen guten Start in die Entwicklung. Dabei sind durch den Einsatz von Technologiebasierten Lösungen aber noch Erweiterungen oder Änderungen zu erwarten. Dabei ist das System in einer 3-Schichten Architektur aufgebaut. Wie in ?? gezeigt nehmen wir eine strikte Trennung zwischen Anwendungslogik und Ressourcenmanagement vor. Das bringt uns den Vorteil das bei Änderungen der Datenhaltung, die Anwendungslogik nicht angepasst werden muss. Als Beispiel sei zu nennen das die Lehrereinrichtung ihre eigene Datenhaltung in das System integrieren möchte.

9.5 Datenstruktur und relevante Informationen

9.5.1 Einleitung

Um im System Informationen verarbeiten zu können, müssen diese erst einmal definiert werden. Im Folgenden möchten wir kurz auflisten, welche Informationen im System eine Rolle spielen und warum. Dabei unterscheiden wir zwischen Benutzerinformationen, Rauminformationen, Filtermöglichkeiten, Belegungspläne und sonstigen relevanten Eigenschaften.

9.5.2 Benutzerinformationen

Eine Verifizierung eines Benutzers des Systems ist notwendig, da gewährleistet werden muss, dass nur Angestellte und Mitglieder der Lehrereinrichtung Zugang zu den System- bzw. Rauminformationen erhalten. Deswegen sollte beim Systemzugriff geprüft werden ob der Benutzer eine Zugangsberechtigung für das System besitzt. Eine einfache Möglichkeit das zu lösen, wäre eine Authentifizierung über eine Lehrereinrichtungskennung, z. B. eine spezielle E-Mail-Adresse. Diese E-Mail-Adresse muss bei der Erstinstallation angegeben werden, und wird bei jeder Interaktion mit dem System übertragen um den Benutzer zu verifizieren. Der Server besitzt einen Datensatz mit berechtigten E-Mail-Adressen, und kann so überprüfen ob die Anfrage des Benutzers bearbeitet werden darf. Aus der E-Mail-Adresse lassen sich dann auch, z. B. über das Format der Adresse oder anhand einer weiteren Liste im System, die Zugangsberechtigungen für bestimmte Räume prüfen. Falls die Lehrereinrichtung nicht möchte das ein bestimmter Raum von jeder Benutzergruppe genutzt werden kann, lässt sich so eine Berechtigungsabfrage durchführen, ohne dass der Benutzer zusätzliche Interaktionsschritte unternehmen muss. Außerdem muss der Standort des Benutzers festgelegt werden damit er durch das System bei der Raumauswahl berücksichtigt werden kann. Zu den infrage kommenden Technologien dazu, kommen wir in einem ??.

9.5.3 Rauminformationen

Um einen Raum im System eindeutig identifizieren zu können, muss die Raumnummer als Ausgangspunkt hinterlegt werden. Zusätzlich muss sämtliches, für das System relevante, Equipment aufgelistet werden, damit im späteren Verlauf eine Suche nach einem bestimmten Raum ermöglicht werden kann. Dieses Equipment kann je nach Raum oder Raum Typ unterschiedlich ausfallen und hängt in der Regel von seinem gedachten Verwendungszweck ab. Räume die für Präsentationszwecke gedacht sind, besitzen in der Regel ein Präsentationsmedium wie etwa Beamer, Fernseher, OHP oder Whiteboard. Zusätzlich werden, wie in fast jedem Raum, Sitzmöglichkeiten benötigt. Die Größe, und damit die Anzahl der Sitzplätze sind ebenfalls relevant, da nur so auf einen Raum für eine bestimmte Anzahl an Personen hingearbeitet werden kann. Die dynamischen Rauminhalte, und mit welchen Technologien diese aktualisiert werden können, werden im Abschnitt [Flexible Räume](Textverweis) behandelt. Diese Rauminhalte sind je nach Lehrereinrichtung und vorhandenem Equipment variabel und müssen sich an dem spezifischen Kontext ausrichten. Eine weitere relevante Information ist der Standort des Raums um einen kurzen Weg zwischen Benutzer und gesuchten Raum zu berechnen. Dieser lässt sich im Idealfall aus der Raumnummer ableiten, indem diese so aufgebaut ist das das Gebäude, Stockwerk und der Gang daraus ersichtlich wird. Auch diese Informationen sind variabel und müssen auf das jeweilige Umfeld angepasst werden. Alternativ lassen sich diese Informationen auch einzeln im System abspeichern.

9.5.4 Filtermöglichkeiten

Damit der Benutzer nach einem Raum suchen kann, muss das System ihm die Möglichkeit bieten auf Wünsche und Möglichkeiten einzugehen. Dazu sollte der Benutzer die Möglichkeit haben durch spezielle Filter seinen Raumwunsch zu konkretisieren. Diese Filter betreffen z. B. den Rauminhalt der benötigt wird damit der Benutzer seine Arbeit verrichten kann. Der Benutzer kann durch auswählen seinen Raumwunsch eingrenzen und dem System mitteilen der diese Wünsche in seiner Raumauswahl berücksichtigt. Der Benutzer sollte aber auch grundlegende Informationen angeben können, wie z. B. den barrierefreien Zugang zu Räumen. Da kann der Benutzer z. B. angeben das er einen Fahrstuhl benötigt um die Etagen im Gebäude wechseln zu können.

9.5.5 Belegungspläne

Da in einer Lehrereinrichtung oft Veranstaltungen existieren die im wöchentlichen Muster wiederholt werden, existiert dazu meist ein Belegungsplan. Dieser gibt an in welchem Raum in welcher Zeitspanne eine Veranstaltung, also eine Belegung des Raumes, stattfindet. Diese Liste mit Informationen, sofern vorhanden, ist für die Ausgabe von freien Räumen an den Benutzer eine wichtige Grundlage. Mit ihr können viele Räume schon vorab ausgeschlossen werden, bevor sie in die Berechnung mit einfließen. Zusätzlich muss eine Liste existieren die Auskunft über die dynamische Belegung von Räumen Aufschluss gibt. In dieser Liste sollten alle aktuellen Belegungen von Räumen verzeichnet sein, damit

das System bei seinen Berechnungen unterstützt wird. Die Liste sollte mit Informationen gefüllt sein die Aufschlüsse über den belegten Raum, die Anzahl der Personen in diesem Raum, Start- und Endzeitpunkt der Belegung sowie die Benutzeridentifizierung des Benutzers, der den Raum gebucht hat, gibt.

9.5.6 Sonstige Eigenschaften

Grundlegende Informationen wie alle zur Verfügung stehenden Räume innerhalb der Lehreinrichtung, die Anzahl der Gebäude, Stockwerke und Gänge, Positionen von Ein-/Ausgängen, Treppenhäuser, Notausgänge, Fahrstühle und eventuell weitere Informationen die vom Umfeld abhängen, sind ebenfalls eine wichtige Grundlage für das System. Mit diesen Informationen und denen der Räume und Benutzer, können im System später Raumvorschläge z.B. für Benutzer die einen barrierefreien Zugang benötigen erstellt werden.

9.5.7 Fazit - Datenstruktur und relevante Informationen

Die benötigten Informationen im System sollten innerhalb eines persistenten Datenspeichers, bspw. einem Datenbankserver hinterlegt sein. Als Datenformat zum senden, empfangen und auswerten von Informationen, haben wir uns für das praktische [?] entschieden. Durch die weite Verbreitung ist es ideal auf verschiedenen Plattformen anwendbar. Die Auflistung an benötigten Informationen und die grobe Datenstruktur befindet sich als ausführliche Beschreibung im ??.

9.6 Standortbestimmung des Benutzers

9.6.1 Einleitung

Damit durch unser System ein Raum ermittelt werden kann, der sowohl den Erfordernissen des Benutzers entspricht, als auch einen möglichst kurzen Laufweg für den Benutzer gewährleistet, muss der Standort des Benutzers innerhalb des Gebäudes bestimmt werden. Als Methoden zur Feststellung des Standortes haben wir verschiedene Möglichkeiten betrachtet.

9.6.2 Standortbestimmung durch Benutzereingaben

Die Standortbestimmung über Benutzereingaben würde dabei so funktionieren, dass der Benutzer einen Bezugspunkt innerhalb eines Gebäudes, z.B. eine Raumnummer oder die Kennzeichnung eines Treppenhauses, angibt. Wir haben uns allerdings gegen diese Methode entschieden da sie eine zusätzliche Interaktion durch den Benutzer erfordert und das gegebenenfalls den Nutzungsfluss des Systems stören könnte. Diese Methode kann allerdings als Fallback verwendet werden, wenn die Standortbestimmung durch andere Methoden oder Technologien scheitert.

- Vorteile:
 - Benutzer kann selber seinen Startpunkt der Suche angeben.
- Nachteile:
 - Benutzereingaben können fehlerhaft sein
 - Benutzer weiß eventuell nicht wo er sich befindet
 - zusätzlicher Interaktionsschritt durch den Benutzer nötig

9.6.3 Standortbestimmung durch GPS

Durch die Verwendung von GPS (Global Positioning System) könnte ebenfalls der Standort des Benutzers bestimmt werden. Dabei würde mittels des Endgerätes des Benutzers die Verbindung zum "Global Positioning System" aufgebaut und somit der Standort mit einer hohen Genauigkeit bestimmt werden. Da die Genauigkeit allerdings schwanken kann haben wir uns gegen die Verwendung von GPS entschieden da eine fehlerhafte oder ungenaue Messung zu falschen Ergebnissen innerhalb des Systems führen kann.

- Vorteile:
 - sehr präzise
 - sehr weit verbreitet
- Nachteile:
 - innerhalb von Gebäuden können Signalstörungen auftreten

9.6.4 Standortbestimmung durch Bluetooth-Beacons

Bluetooth-Beacons sind das Indoor-Äquivalent zum klassischen GPS und ermöglichen innerhalb von Gebäuden eine exakte Standortbestimmung auf bis zu 1 Meter [INFSOFT GMBH]. Dazu kann eine Entfernungsbestimmung vom Endgerät des Benutzers zum Beacon durch Auslesen von Signaldaten aufgestellt werden, womit ein grober Standort bestimmt werden kann. Durch die Bluetooth Low Energy Technik (BLE) die seit Version 4.0 im Bluetooth Industriestandard spezifiziert wurde, sind diese Beacons äußerst stromsparend und können dabei im Batteriebetrieb je nach eingestelltem Sendeintervall 2-8 Jahre [INFSOFT GMBH] senden. Durch die Möglichkeit diese an das Stromnetz anzuschließen, lässt sich die Betriebsdauer beliebig erhöhen. Auf dem Markt sind bereits spezielle Beacons vorhanden die für das Bestimmen von Standorten wie z.B. Messegeländen oder Flughafenterminals konzipiert worden sind. Estimote, Inc. [?] bietet eine spezielle Lösung zu Standortbestimmung innerhalb von Gebäuden an. Dabei lässt sich mit Hilfe eines Tools eine Karte eines Gebäudes erstellen, auf der durch geschicktes platzieren von speziellen Estimote Beacons [ESTIMOTE, INC., 2017] der Standort

des Benutzers auf einer Karte [ESTIMOTE, INC., 2017] angezeigt werden kann. Allerdings ist diese Funktion nur exklusiv bei Estimote, Inc. [ESTIMOTE, INC., 2017] verfügbar, weswegen sie im Rahmen unseres Projektes nicht anwendbar ist. Allerdings lässt sich ein Standort auch ohne interaktive Karte bestimmen, da dafür nur normale Beacons benötigt werden. Die Open Source Projekt Android Beacon Library [?] von Radius Networks [?] biete eine Anbindung an viele verschiedene Beacon Modelle und Marken die dem AltBeacon standard[?] entsprechen. Über z. B. eine Android Applikation lassen sich verschiedene Informationen von einem Beacon empfangen und auslesen.

- Vorteile:
 - günstig
 - einfach zu handhaben
 - bekannte, altbewährte Technologie (Bluetooth)
 - Langlebigkeit kann beliebig erhöht werden
- Nachteile:
 - Bluetooth und Wifi-Signale teilen sich eine Frequenz (2,4 GHz), kann zu Störungen führen
 - Signaldämpfung durch Wasser und Metall im Sendebereich ist sehr hoch

9.6.5 Fazit - Standortbestimmung des Benutzers

Nach unserer Recherche, wäre das Verwenden von Beacons zur Standortbestimmung ideal für unser Projekt geeignet. Durch die günstigen Anschaffungskosten und die Langlebigkeit der einzelnen Beacons lassen sie sich auch in großen Massen innerhalb eines Gebäudes anbringen. Dass die BLE Beacons sowieso schon für die Standortbestimmung eingesetzt werden, unterstützt unsere Ansicht. Nach unseren Erkenntnissen wäre das Anbringen eines Beacons an jedem Raum bzw. in einem bestimmten, festgelegten Abstand von Vorteil, da dadurch eine Beacon-Raum Zuweisung erreicht werden kann. Anhand dieser Informationen, die im System hinterlegt sind, kann eine Standortbestimmung durch den Benutzer erfolgen. Das Endgerät des Benutzers scannt dabei selbstständig nach Beacons in der Nähe und speichert diese. Wenn der Benutzer einen freien Raum sucht, kann der zuletzt gespeicherte Beacon bzw. seine ID, im Request mit übergeben werden. Damit kann das System den Standort des Benutzers anhand des erkannten Raumes in seiner Nähe in seine Berechnung für einen Raum mit einbeziehen. Zusätzlich kann durch das Auslesen von Entfernungen zwischen Beacon und Benutzer eine genauere Standortbestimmung erfolgen. Dafür muss das System die Position aller Beacons in Abhängigkeit zu ihrem zugehörigen Raum wissen. Um diese Zuordnung einfach zu halten, wäre es ratsam die Beacons direkt an oder über der Tür zum Raum anzubringen. So kann erreicht werden das jede Person im Gang das Signal eines Beacons empfangen kann. Damit Signaldämpfungen minimiert werden, sollten die Beacons in einer Höhe angebracht werden, die über den Köpfen der Personen liegt. Dafür würde sich der Bereich über der Tür, in ca. 2 Meter Höhe anbieten. Um große Abstände zwischen 2 Räumen

zu überbrücken, sollte in einem festgelegten Abstand ein Beacon angebracht werden, der im System als Pseudo-Raum zur Positionsbestimmung genutzt werden kann. Damit ein kurzer Weg zwischen Benutzer und gewünschten Raum errechnet werden kann, müssen alle möglichen Laufwege im System vermerkt sein. Dazu gehören auch markante Punkte wie Treppenhäuser, Ein-/Ausgänge und Fahrstühle die ebenfalls mit einem Beacon ausgestattet werden sollten, um die Standortbestimmung des Benutzers zu optimieren. Für den Fall das kein Beacon durch das Endgerät des Benutzers erkannt werden kann, sollte, sofern vorhanden, der letzte erkannte Beacon als aktueller Standort genutzt werden. Dabei sollte der Benutzer darauf hingewiesen werden das kein Beacon im Umkreis gefunden wurde und deshalb eine korrekte Standortbestimmung nicht verfügbar ist. Wahlweise sollte der Benutzer die Möglichkeit haben einen Startpunkt zu wählen, bspw. der Haupteingang des Gebäudes. Der Einsatz von Bluetooth Beacons birgt auch einige Risiken, die wir im ?? dokumentiert haben. Als Ergänzung ist bei jedem Risiko angegeben wie wir mit ihm verfahren werden sofern es eintritt. Die automatische Standortbestimmung des Benutzers ist Teil unserer Anwendungslogik, da ohne sie die angestrebten Ziele nicht erreicht werden können. Die suche nach Beacons erfolgt automatisch sobald die Anwendung einmal gestartet wurde. Das Anwendungsobjekt das durch die Anwendungslogik verarbeitet wird ist in diesem Fall eine Liste mit Beacon ID's die in der Umgebung des Benutzers gefunden wurden. Zusätzlich wird die Liste nach dem nächstgelegenen Beacon sortiert um bei einer Anfrage an den Server diese Beacon ID mit zu übermitteln.

9.7 Laufwegoptimierung

9.7.1 Einleitung

Das deskriptive Aufgabenmodell der Raumsuche im Kontext der Lehrereinrichtungen, setzt das manuelle suchen von Räumen innerhalb des Gebäudes voraus. An diesem Punkt kann viel Zeit verschwendet werden, wenn sich zusätzliche Laufwege von Raum zu Raum ergeben, da festgestellt wird das ein gerade überprüfter Raum schon belegt ist oder nicht das Equipment beinhaltet das vom Benutzer benötigt wird. Das Problem hängt eng mit der Standortbestimmung des Benutzers zusammen, da ein Laufweg immer von der aktuellen Position des Benutzers zum, vom System berechneten, nächsten freien Raum der den Bedürfnissen und Erwartungen des Benutzers entspricht berechnet werden muss. Im Folgenden wollen wir dieses Problem, das unserem Alleinstellungsmerkmal entspricht, versuchen zu lösen.

9.7.2 Verkettete Liste / Dijkstra-Algorithmus

Damit ein Weg berechnet werden kann, werden in unserem Fall mindestens 2 Punkte benötigt. Zum einen der aktuelle Standort des Benutzers den wir über Bluetooth Beacons ermitteln werden, und zum anderen der Raum bspw. die Raumnummer des Raumes den der Benutzer benötigt. Um einen optimalen Weg berechnen zu können muss das System Abwägungen über mehrere Räume aber auch mehrere vorhandenen Laufwegen durchführen können. Wir sind zu dem Entschluss gekommen das die

einfachste Methode eine verkettete Liste wäre die im System hinterlegt ist und anhand eines leicht modifizierten Dijkstra-Algorithmus die Raumsuche übernimmt. Dabei muss für jede Lehreinrichtung eine eigene Liste erzeugt werden die sich aus der Anzahl der Räume, ihren Standorten und Rauminhalten sowie besonderen Merkmalen innerhalb des Gebäudes zusammensetzt. Dabei sollte z. B. der Administrator des Systems der Lehreinrichtung, die Möglichkeit haben diese Liste jederzeit zu aktualisieren um auf z. B. Neuanschaffungen oder Umfunktionierung von Gebäuden oder Räumen eingehen zu können. Jeder Raum im System wird als ein Knoten innerhalb der Liste dargestellt. Dabei sind angrenzende Knoten in der Liste gleichbedeutend mit angrenzenden Räumen im Gebäude. So lässt sich im System ein Weg von Raum zu Raum bzw. von Knoten zu Knoten berechnen. Wie schon erwähnt, sollten auch markante Punkte wie Treppenhäuser oder Fahrstühle als Knoten innerhalb der Liste angegeben werden, da sie ebenfalls als Wegstrecke zwischen zwei Punkten genutzt werden können. Als Ergebnis enthält man eine Liste in der alle Räume und markante Punkte des Gebäudes als Knoten modelliert wurden. Die einzelnen Knoten besitzen verschiedene Attribute die diesen Knoten beschreiben. Jeder Knoten enthält eine eindeutige ID die ihn innerhalb der verketteten Liste eindeutig identifiziert. Die ID lässt sich dabei frei bestimmen und sollte dementsprechend den Typ des Knoten kennzeichnen. Bspw. sollten Raumknoten mit dem Präfix "room" gekennzeichnet werden, wobei ein Treppenhaus das Präfix "stairs" beinhalten sollte. Das sorgt dafür das ein Knoten schneller und gezielter angesprochen werden kann. Zusätzlich sollte die ID die, in der Regel bereits bestehende, Raumnummer beinhalten. Existiert keine Raumnummer, muss der Raum im System mit einer eindeutigen Kennung versehen werden, durch die der Raum sich identifizieren lässt. Dabei sollte dann aber auch berücksichtigt werden das der Benutzer eine Raumnummer oder eine ähnliche Kennzeichnung benötigt um den Raum im Gebäude zu finden. Hierbei muss auf das gegebene Umfeld Rücksicht genommen werden. Ist jeder Raum im System eindeutig identifiziert und über einen Namen oder ID in der verketteten Liste vorhanden, sollte jeder Knoten zusätzliche Eigenschaften erhalten die für die Berechnung eines Raumes für den Benutzer benötigt werden. Damit ein Raumvorschlag ausgegeben werden kann, muss das System den kürzesten Weg zwischen Benutzer und möglichen Räumen berechnen. Dabei muss das System eine Entscheidung treffen können welcher der infrage kommenden Räume er dem Benutzer vorschlagen sollte um einen möglichst kurzen Weg zu gewährleisten. Um diesen Weg zu berechnen haben wir uns eine Gewichtung überlegt anhand dessen das System den kürzesten Weg zwischen 2 Punkten berechnen kann. Jeder Knoten besitzt die Gewichtung zu jedem seiner angrenzenden Knoten die vom Algorithmus ausgelesen wird. Wir treffen dabei die Unterscheidung zwischen Verbindungen zu Räumen, Treppenhäusern oder Aufzügen. In ?? ist so eine Gewichtung schematisch zum besseren Verständnis abgebildet. Dabei ist ersichtlich das unterschiedliche Knotentypen eine unterschiedliche maximale Anzahl an Kanten besitzen kann. Wir haben uns auf folgende maximale Kantenzahl festgelegt:

- Räume: 1-2 benachbarte Knoten/genutzte Kanten
- Fahrstühle/Treppenhäuser: 1-6 benachbarte Knoten/genutzte Kanten (Norden, Süden, Westen, Osten, Oben, Unten)
- Ein-/Ausgänge: 1-6 benachbarte Knoten/genutzte Kanten

Um den Algorithmus sinnvoll anwenden zu können, sollte für jedes vorhandene Gebäude mit systemrelevanten Räumen eine eigene verkettete Liste angelegt werden. Diese Listen sollten über Einstiegs- bzw. Austrittsknoten logisch miteinander verbunden werden. Dies ist nötig um zu gewährleisten das ein Ausgang eines Gebäudes zu beliebig vielen Eingängen anderer Gebäude führen kann. Der von uns definierte Algorithmus sucht sich vom Standpunkt des Benutzers ausgehend den Weg, der am kürzesten ist. Dabei wird immer die niedrigste Summe der Kantengewichtungen fokussiert betrachtet. Ergibt sich auf einem anderen Weg eine niedrigere Summe als auf dem bisherigen Weg, wird dieser Weg in den Fokus der Raumermittlung gelegt. Der erste Raum den der Algorithmus erreicht wird als Raumvorschlag an den Benutzer ausgegeben. Um nicht jeden einzelnen Weg berechnen zu müssen wird bei der Berechnung in zwei Arten unterschieden.

1.0.0 Der Benutzer und der Zielraum befinden sich in unterschiedlichen Gebäuden

- In diesem Fall wird als Zwischenpunkt der Weg zu einer Übergangsmöglichkeit gesucht. Es wird z. B. der nächstgelegene Ausgang aus dem aktuellen Gebäude gesucht. Von dort aus wird die Kante zum Zielgebäude gewählt, in dem sich der Zielraum befindet.

2.0.0 Der Benutzer und der Zielraum befinden sich im selben Gebäude

- Trifft das zu, wird der normale Algorithmus angewandt, in dem bei allen Knoten bis hin zum Zielraum geprüft wird in welcher Entfernung er sich befindet.

Diese Unterscheidung hilft uns dabei unnötige Berechnungszeiten zu vermeiden und effektiver Anfragen bearbeiten zu können.

Mit diesen Berechnungsmethoden für freie Räume in der Nähe des Benutzers kann keine metergenaue Angabe gemacht werden, die Entfernung bezieht sich lediglich auf die angegebene Gewichtung jedes Knoten. Deswegen ist eine passende Gewichtungsangabe dringend erforderlich. Es gilt je nach Gebäude und Abstände der Räume untereinander eine unterschiedliche Gewichtung zu legen damit eine möglichst realistische Wegberechnung aufgestellt werden kann. Diese Gewichtung muss je nach Anwendungsumgebung individuell festgelegt werden. Zum Beispiel müssen Stellen berücksichtigt werden wo zwischen 2 Räumen eine größere Entfernung vorhanden sind als normal. Die Gewichtung muss zwischen diesen 2 Knoten also erhöht werden um die erhöhte Entfernung zu simulieren. Ein Stockwerkwechsel sollte ebenfalls mit einer speziellen Gewichtung angegeben werden. Ein Treppenhaus bedeutet einen zusätzlichen Aufwand und je nach Umgebung auch einen erhöhten Zeitbedarf der in Form einer erhöhten Gewichtung berücksichtigt werden muss. Für den Spezialfall das ein Benutzer einen barrierefreien Zugang benötigt und Räume infrage kommen die sich in einem anderen Stockwerk befinden, wird als Zwischenziel ein Fahrstuhl oder andere barrierefreie Möglichkeit zum Stockwerkwechsel gesucht.

9.7.3 Fazit - Laufwegoptimierung

Die verkettete Liste mit einem Dijkstra-Algorithmus ergibt in unserem Projekt eine sinnvolle Methode um die kürzeste Entfernung zwischen einem Start- und mehreren Endpunkten zu bestimmen. Das der

Einsatz einer verketteten Liste in diesem Anwendungsumfeld funktioniert, zeigt die Durchführung des Proof of Concepts den wir im ?? dokumentiert haben. Durch eine Anwendungslogik wird regelmäßig automatisch eine Liste mit aktuell freien Räumen erstellt auf die der Algorithmus zurückgreift. Dabei verzichten wir extra auf eine vorab Berechnung aller Routen zwischen jedem Knoten, weil damit keine dynamische Gewichtsänderung im laufenden Betrieb durch z.B. sperren von Gängen oder Ein-/Ausgängen berücksichtigt werden kann.

9.8 Flexible Räume (Verteilte Anwendungslogik)

9.8.1 Einleitung

Das innerhalb von Räumen befindliche Equipment ist bis auf wenige Ausnahmen in der Regel nicht fest installiert. So hat der Benutzer die Möglichkeit Gegenstände von Raum zu Raum zu tragen um Räume flexibel nutzen zu können. Im System müssen diese Verschiebungen von Equipment durch irgendeine Methode vermerkt werden um zu gewährleisten das immer bekannt ist was sich in einem Raum gerade an Ausrüstungen befindet. Im Folgenden sollen Möglichkeiten diskutiert werden, die zur flexiblen Raumgestaltung eingesetzt werden können.

9.8.2 Aktives aktualisieren von Rauminhalten

Zur Lösung dieses Problems muss innerhalb des Systems eine Erkennung erfolgen die die dynamische Equipment Verschiebung erkennen und im System vermerken kann. Die einfachste Möglichkeit wäre durch das Einbinden des Benutzers möglich. Dabei hat der Benutzer die Aufgabe durch eine Interaktion mit dem System Gegenstände die er aus einem Raum entfernt, aus dem Inventar des Raumes auszutragen. Im nächsten Schritt muss er das entwendete Equipment im gewünschten Raum zum Inventar hinzufügen. Die Umsetzung könnte innerhalb eines einfachen Dialogs zwischen Benutzer und System erfolgen indem er den zu entfernenden Gegenstand des Raumes auswählt und durch z. B. einen Button dem System mitteilt das dieser Gegenstand nicht länger Teil des Inventars des Raumes ist. Im nächsten Schritt trägt der Benutzer den Gegenstand im Inventar des Zielraums ein. Um zu gewährleisten das der Gegenstand im gewünschten Zielraum auch wieder eingetragen wird sollte ein durchgehender Prozess dafür sorgen das der Benutzer nach dem Austragen eines Gegenstandes aus einem Raum als nächsten Schritt zwingend das erneute eintragen des Gegenstandes in einem Raum vornehmen muss. Andernfalls besteht die Gefahr das ein Gegenstand aus dem System verschwindet und nicht mehr auffindbar ist. Die Möglichkeit den Benutzer für diese Aufgabe einzubeziehen halten wir allerdings nicht für optimal, da der Benutzer zusätzliche Interaktionsschritte unternehmen muss um dieses Ziel zu erreichen. Eine automatisierte Möglichkeit ohne den Benutzer zusätzlich zu belasten wäre zu bevorzugen.

9.8.3 Passives aktualisieren von Rauminhalten

Nach einer Recherche zum Thema Gegenstandserkennung bzw. Zuordnung sind wir auf die RFID-Technologie (Radio Frequency Identification) gestoßen. Durch das Markieren von Gegenständen und dem Anbringen eines Lesegeräts lassen sich Gegenstände über die RFID-Technik erkennen. Dabei kommunizieren die beiden Komponenten über elektromagnetische Wellen miteinander, die in der Regel vom Lesegerät ausgesandt, und über kleine Antennen empfangen werden. Gleichzeitig wird bei manchen RFID-Transpondern der benötigte Strom zum Senden von Informationen über diese elektromagnetischen Wellen vom Lesegerät mit übertragen. Das bringt den Vorteil, dass die RFID-Transponder keine eigene Stromquelle benötigen und somit passiv kommunizieren können. Beim Betreten des Empfangsbereiches des Lesegerätes wird sowohl der benötigte Strom übertragen, als auch Daten an das Lesegerät gesendet. Für unseren Kontext sind die passiven RFID-Transponder ideal, da eine Vielzahl an Gegenständen mit diesen Transpondern ausgestattet werden müssen. Um dieses Prinzip umzusetzen, muss jeder Gegenstand, der relevant für das System ist, einen Transponder und eine eindeutige ID besitzen, die ihn im System identifiziert. Es gibt verschiedene Transponder, die für unterschiedliche Einsatzzwecke geeignet sind. In unserem Fall wären flache, selbstklebende Transponder ein Vorteil, da diese schnell und einfach am entsprechenden Gegenstand angebracht werden können. Diese Labels [?] kommen auch oft in der Versand- und Logistikabteilung zur Anwendung, um Postsendungen oder Paletten zu kennzeichnen. Das Prinzip lässt sich auch auf unser Projekt übertragen. Um mutwilliger oder unabsichtlicher Zerstörung dieser Labels zu verhindern, sollten diese an Positionen angebracht werden, die möglichst nicht ersichtlich sind. Die Gefahr, dass ein Label beschädigt und dadurch nicht mehr erkannt wird, besteht allerdings weiterhin. Die Auswahl des entsprechenden Transponders und Lesegerätes hängt stark vom Anwendungsgebiet und dem Umfeld ab. Unterschiedliche Frequenzbereiche können unterschiedlich große Reichweiten gewährleisten. Diese reichen von einem Zentimeter bis hin zu mehreren Metern. Da in unserem Fall das Lesegerät bzw. die Antenne im Türbereich angebracht werden soll, wird eine Empfangsreichweite von ca. einem Meter benötigt. Diese Entfernung würde man in einem *Ultra hohen Frequenzbereich* (UHF) erreichen. Diese sendet zwischen 860 und 960 MHz [?] und erkennt selbst passive Transponder über mehrere Meter. Die UHF-RFID-Tags sind dabei papierdünn, günstig und können nahezu überall angebracht werden. Um unabsichtliches Einlesen von RFID-Transpondern im Türbereich zu vermeiden, sollte die Antenne so ausgerichtet werden, dass sie nur von einem Türrahmen zum gegenüberliegenden Türrahmen empfangen kann. Das Anbringen des Lesegerätes bzw. der Antenne am Türrahmen selber wäre allerdings keine gute Idee, da es möglich ist, dass Personen beim Passieren der Tür die Antenne berühren oder sogar beschädigen. Unserer Meinung nach sollte diese Antenne direkt neben dem Türrahmen wie in ?? gezeigt positioniert werden. Um den Empfangsbereich der Antenne weiter einzugrenzen und versehentliches Einlesen zu vermeiden, wäre das Anbringen eines kleinen Reflektors um die Antenne herum von Vorteil.

Um die erkannten Gegenstände bzw. RFID-Transponder im System zu erkennen und Aktualisierungen zu speichern, wird als zusätzliche Komponente eine Verarbeitungseinheit, z. B. ein Raspberry Pi 3, benötigt. Das Lesegerät empfängt dabei die erkannten Gegenstände im Empfangsbereich und sendet diese über ein Datenkabel an den Raspberry Pi. Dieser hat in einer Liste alle Rauminformationen

inklusive aktuellem Equipment gespeichert und kann so automatisch auf Änderungen reagieren. Wird vom Lesegerät ein Gegenstand erkannt, wird die ID des Gegenstandes ausgelesen und an den Raspberry Pi gesendet. Dieser gleicht die ID mit seiner vorhandenen Liste an Equipment ab. Ist die ID in der Liste vorhanden, also der Gegenstand ein Teil des Inventars, trägt die Anwendung auf dem Minicomputer diesen Gegenstand aus der Inventarliste aus. Zeitgleich wird mit der PUT-Methode ein HTTP Request an den Server gesendet um ihn über die Aktualisierung zu informieren. Der Server liest den Body des Requests aus in dem die Änderung (Gegenstand ausgetragen/eingetragen) und die ID des entsprechenden Gegenstandes vermerkt sind. Der entsprechende Gegenstand wird im Datensatz wo alle Gegenstände und ihre aktuelle Position vermerkt sind ausgetragen. Um zu vermeiden das ein Gegenstand nach dem Austragen im System nicht mehr vorhanden ist, wird die ID des Gegenstandes mit dem Zeitpunkt des Austragens in eine spezielle Liste geschrieben. Auf dieser bleibt er solange vorhanden, bis er in einem anderen Raum erkannt wird. Somit ist gewährleistet das ein Gegenstand immer im System bekannt ist. Das dient uns als zusätzliche Überprüfung über den Verbleib des Gegenstandes. Der Administrator des Systems kann so regelmäßig prüfen ob Gegenstände unterwegs abhandengekommen sind und Nachforschungen über den Verbleib anstellen.

Erkennt das Lesegerät eines Raumes einen Gegenstand der nicht Teil des aktuellen Rauminventars ist, weiß der Minicomputer das dieser Gegenstand den Raum gerade betreten hat. Bevor er den Gegenstand aber in seiner Inventarliste vermerkt, fragt er den Status des Gegenstandes im System ab. In Fehlerfällen kann es möglich sein das ein Gegenstand versehentlich Teil eines Rauminventars geworden ist, obwohl das nicht beabsichtigt war. Wird ein Gegenstand zu nahe an das Lesegerät eines Raumes getragen kann er versehentlich dort in das Inventar eingetragen werden. Das kann passieren, wenn der Gegenstand nur bis in den Empfangsbereich, aber nicht durch diesen hindurch getragen wird. Er wird also nur einmal vom Lesegerät erkannt und dann in das Inventar des Raumes eingetragen. Verlässt der Gegenstand den Empfangsbereich in Richtung Ausgang, kann er vom System nicht als Gegenstand erkannt werden der den Raum verlässt. Wird der Gegenstand dann in den eigentlichen Raum getragen den der Benutzer beabsichtigt, würde er vom Lesegerät erkannt und ebenfalls im Rauminventar des zweiten Raumes eingetragen. Damit verhindert wird das ein Gegenstand im System in 2 Räumen gleichzeitig vorhanden ist, muss eine Überprüfung erfolgen. Wenn der Gegenstand bereits Teil eines anderen Rauminventars ist, muss er aus diesem ausgetragen und in das Inventar des zweiten, neuen Raumes eingetragen werden. Dazu sendet der Server dem ersten Minicomputer über einen PUT-HTTP-Request den Befehl zum Austragen aus seinem Inventar und nach Bestätigung im Response dem 2. Minicomputer die Erlaubnis den erkannten Gegenstand in sein Inventar einzutragen. Den Minicomputer als Verarbeitungseinheit haben wir im ?? und ?? ergänzt.

9.8.4 Fazit - Flexible Räume (Verteilte Anwendungslogik)

Die Überprüfung ob der Einsatz von RFID Technik für die Equipmentmarkierung in unserem Anwendungsfeld funktioniert simulieren wir durch erstellte Datensätze. Dabei können wir aufgrund von mangelnder Technologie aktuell nicht auf echte RFID Transponder und Lesegeräte zurückgreifen. Um diese Technik dennoch zu prüfen haben wir eine Simulation erstellt die überprüft ob der vorgesehene

Ablauf funktioniert. Die Dokumentation des Proof of Concepts befindet sich im ???. Das konkrete Anwendungsobjekt das über Anwendungslogik verarbeitet wird, ist die Liste mit Informationen über den aktuellen Raumbestand eines Raumes. Verändert sich die Liste durch ein- bzw. austragen von Gegenständen, wird die Liste des Raumes auf dem Minicomputer aktualisiert. Diese aktualisierte Liste wird dann über HTTP-Request an den Server gesendet, der diese neuen Informationen verarbeitet und in der Datenbank in aktualisierter Form speichert.

9.9 Verschiedene Eigenschaften der Räume

9.9.1 Einleitung

Um im Ablauf des Systems zu gewährleisten das ein Raum auch tatsächlich frei ist wenn der Benutzer ihn erreicht, wird eine Methode benötigt die dieses Problem lösen kann. Im Folgenden beschäftigen wir uns mit der Zugangsberechtigung zu einem Raum und wie der Benutzer Zugang zu diesem bekommt. Außerdem beschäftigen wir uns mit unterschiedlichen Raumtypen.

9.9.2 Zugang zum Raum

Um gewährleisten zu können das ein Raum frei ist wenn der Benutzer ihn erreicht, muss eine Zugangsberechtigung eingerichtet werden. Bei offenen Räumen besteht die Gefahr das Personen diesen einfach belegen ohne das System zu nutzen. Der Raum wird im System als frei angezeigt obwohl das nicht der Fall ist. Um das Problem zu lösen wäre das generelle Abschließen aller Räume eine Möglichkeit. Um Zugang zum Raum zu bekommen muss das System diesen gewähren indem geprüft wird ob ein Benutzer eine aktive Reservierung für diesen Raum besitzt. Unserer Idee nach kann das öffnen über einen vom System generierten Code erfolgen der bei der Reservierung an das Endgerät des Benutzers gesendet wird. Um die Tür zu öffnen muss z. B. dem Minicomputer den wir in ??? als neue Komponente des Systems identifiziert haben, dieser Code mitgeteilt werden. Stimmt der Code mit dem im System hinterlegten Raum überein, kann die Tür geöffnet werden. Um das zu realisieren muss das System sowohl dem Benutzer als auch dem Minicomputer innerhalb des Raumes diesen Code über HTTP-Requests mitteilen. Damit Benutzer und Minicomputer im Raum miteinander kommunizieren können haben wir Bluetooth und NFC als Möglichkeiten in Betracht gezogen. Bluetooth, wie schon bei der Standortbestimmung des Benutzers verwendet, ist eine weit verbreitete Datenübertragungstechnik die über kurze Distanz gut funktioniert. Der im Endgerät des Benutzers gespeicherte Code wird dem Minicomputer im Raum über kurze Distanz mitgeteilt. Der Benutzer muss sich dabei unmittelbar vor der Tür befinden damit der Minicomputer den Code empfangen kann. Um auszuschließen das der Code von fremden Personen abgehört und missbraucht wird, sollte der Code im System dynamisch und zufällig generiert werden.

Für zusätzliche Sicherheit sollte erst durch eine Bestätigung des Benutzers dieser Code per Bluetooth an den Minicomputer übertragen werden. Damit der Minicomputer das Bluetooth Signal empfangen kann muss das Endgerät des Benutzers als sendender Beacon umfunktioniert werden. Durch Bluetooth

4.0 und die Einführung der Low Energy Technik muss keine direkte Verbindung mehr hergestellt werden was uns hier zugute kommt ?. Eine der größeren Änderungen bei dem automatischen Entsperren der Räume ist die Notwendigkeit das ein Computer das öffnen übernimmt. Dementsprechend muss der Schließmechanismus so umgerüstet werden das dies automatisch erfolgen kann. Das kann mit elektronischen Schließzylindern ermöglicht werden die in vielen Varianten angeboten werden. In unserem Fall wäre eine drahtlose Kommunikation zwischen Minicomputer und Schließsystem von Vorteil. Diese lässt sich ebenfalls über Bluetooth regeln. NFC als Alternative zu Bluetooth funktioniert aus der Perspektive Benutzers genauso. Allerdings muss der Benutzer sein Endgerät auf ein NFC-Sensorfeld legen da die Kommunikation nur über wenige Zentimeter funktioniert. Die nicht-automatisierte Alternative wäre zu akzeptieren das bei unverschlossenen Räumen Personen den Raum einfach besetzen ohne das System zu nutzen. Wie die Leereinrichtung mit diesem Problem intern umgeht ist ihr selber überlassen.

9.9.3 Raumtyp 'Stiller Arbeitsraum'

Da wir in unserem System einen besonderen Raum haben in denen mehrere verschiedene Benutzer gleichzeitig und alleine arbeiten können, kann für diesen Raumtyp die bisherige Lösung zur Raumbuchung so nicht angewandt werden. Die besondere Eigenschaft an diesem Raumtyp bezieht sich darauf das die Benutzer in der Regel zu unterschiedlichen Zeiten anfangen zu arbeiten und somit eine einzige Buchungszeit kontraproduktiv wäre. Um das Problem zu lösen sind wir zu dem Entschluss gekommen das dieser Raum ein besonderes Datenschema benötigt. Ein Raum der zum stillen arbeiten genutzt wird, hat eine bestimmte Slotzahl an Arbeitsplätzen zur Verfügung in die Benutzer eingetragen werden können. Bucht ein Benutzer einen Raum zum stillen arbeiten, wird er mit dem Zeitstempel seiner Buchung im System vermerkt. Der Raum wird wieder komplett freigegeben wenn der letzte Benutzer den Raum nicht weiter benutzt. So kann zusätzlich eine sinnvolle Raumbesetzung erreicht werden, da dynamisch Räume zum stillen arbeiten erzeugt und wieder entfernt werden.

9.9.4 Fazit - Verschiedene Eigenschaften der Räume

Das Identifizieren und verarbeiten der Probleme SZugang zum Raum und dem SStillen Arbeitsraum als besonderer Raumtyp haben uns dabei geholfen mehr Leereinrichtungen mit diesem Projekt adressieren zu können.

9.10 Personenerkennung im Raum

9.10.1 Einleitung

Zur Findung einer Methode für effektive Personenzählung haben wir ein Brainstorming durchgeführt verschiedene Methoden betrachtet. Dazu gehören Methoden wie die algorithmische Personenzählung durch Wärmebild- und Videokamera, aber auch die Erkennung durch Lichtschranken, Druckplatten

und Bewegungsmelder. Zur Übersichtlichkeit wird in diesem Teil der Dokumentation nur kurz auf die einzelnen Vor- und Nachteile eingegangen. Eine ausführliche Analyse der identifizierten Technologien befindet sich im ??.

9.10.2 Druckplatten im Boden

Bei der Zählung von Personen durch eine druckempfindliche Matte im Eingangsbereich kann durch Algorithmen sowohl die Anzahl der Personen als auch die Laufrichtung dieser durch Analyse der Drucksensoren bestimmt werden.

- Vorteile:
 - unauffällige Personenzählung
 - Laufrichtung erkennbar
 - robust
- Nachteile:
 - Gegenstände könnten als Personen erkannt werden

9.10.3 Lichtschranken und Bewegungsmelder

Lichtschranken oder Bewegungsmelder im Eingangsbereich können dafür genutzt werden um eine Personenanzahl im Raum zu ermitteln.

- Vorteile:
 - keine Individuenerkennung
 - Laufrichtung erkennbar
- Nachteile:
 - ungenau bei mehreren Personen gleichzeitig im Empfangsbereich
 - keine Unterscheidung zwischen Mensch und Gegenstand

9.10.4 Wärmebildkameras

Eine effektive Methode welche auch die persönlichen Daten der Benutzer schützen würde wäre die Verwendung einer Wärmebildkamera mit zugehörigem Algorithmus zur Personenzählung.

- Vorteile:
 - keine Individuenerkennung
 - hohe Genauigkeit
- Nachteile:
 - sehr teuer in der Anschaffung
 - Personen können durch Objekte verdeckt werden

9.10.5 Video-/Bildanalyse

Eine ähnliche und kostengünstigere Alternative bietet eine Analyse von Video/- bzw. Bildmaterial mit der durch algorithmische Personenerkennung die Anzahl der Personen ermittelt werden kann.

- Vorteile:
 - hohe Genauigkeit
 - einfach in der Implementation
- Nachteile:
 - Personen können identifiziert werden, kein Datenschutz gewährleistet
 - Personen können durch Objekte verdeckt werden
 - für Videoanalyse wird ein starker Server benötigt
 - hochauflösende Videos können nur langsam verarbeitet werden

9.10.6 NFC

Eine weitere Möglichkeit der Benutzererkennung innerhalb von Räumen wären der Einsatz von NFC (Near Field Communication). Fast jedes neue Smartphone besitzt die Möglichkeit NFC einzusetzen. Bekannt ist diese Methode z. B. durch das kontaktlose Bezahlen mit einem Smartphone.

- Vorteile:
 - kontaktlose Identifikation
- Nachteile:
 - Reichweite stark begrenzt
 - erfordert in der Regel Benutzerinteraktion

9.10.7 BLE Beacons

Der Einsatz von Beacons wurde bereits bei der Standortbestimmung von Benutzern angesprochen. Ebenfalls anwendbar wäre diese Methode bei der Bestimmung von Benutzern innerhalb von Räumen. Es wird also nach dem Standort innerhalb eines bestimmten Bereiches gesucht.

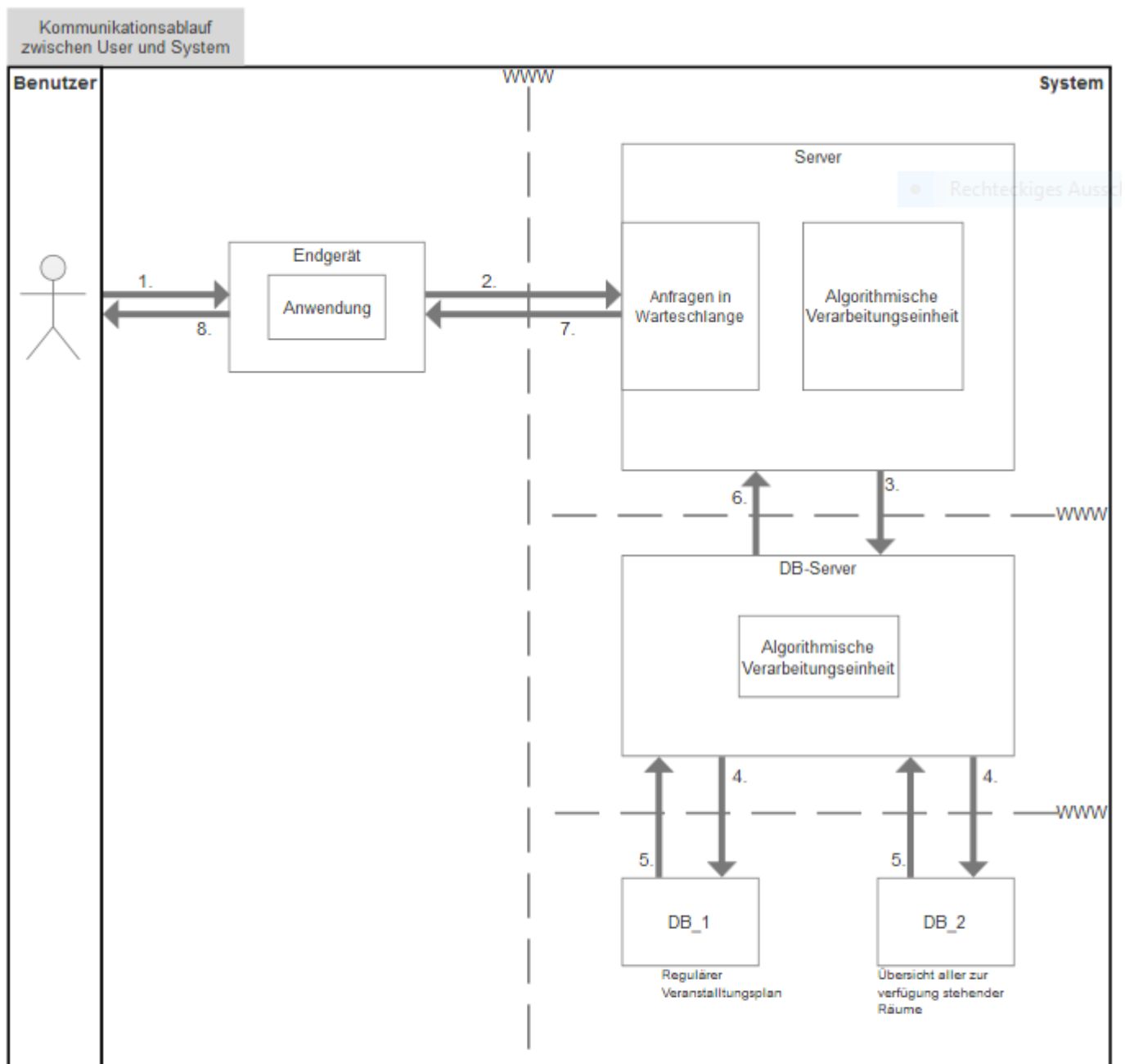
- Vorteile:
 - basierend auf weit verbreitetem Protokoll
- Nachteile:
 - setzt Empfangsgerät beim Benutzer voraus

9.10.8 Fazit - Personenerkennung im Raum

Nach Abwägung der Vor- und Nachteile der einzelnen Technologien sind wir zu dem Entschluss gekommen das für dieses Projekt die Personenerkennung durch Bildanalyse die beste Methode ist. Sie bietet eine ausreichende Genauigkeit sofern die Kamera im Raum richtig positioniert wurde. Der Datenschutz ist ebenfalls gewährleistet wenn die aufgenommenen Bilder ausschließlich innerhalb des Systems verarbeitet werden. Dazu würde sich der Minicomputer innerhalb des Raumes anbieten. Da Personenerkennung in Videos und Bildern ein aktuelles Thema ist, kann man davon ausgehen das in Zukunft bestehende Algorithmen verbessert oder neue Algorithmen entwickelt werden.

9.11 Fazit

In unserem Projekt haben wir uns mit der Umsetzung eines Systems beschäftigt, dass dem Benutzer Zeit sparen soll wenn er innerhalb einer Lehreinrichtung einen Raum zum arbeiten sucht. Wir haben uns bei der Konzeption und Entwicklung des Systems mit dem Vorgehensmodell des [?] beschäftigt, User Roles und Use Cases erarbeitet und diese mit Hilfe zweier Studenten iterativ überarbeitet. Die User Interfaces für die Benutzer wurden mit Hilfe der 5 Regeln und 6 Prinzipien der Gebrauchstauglichkeit erarbeitet und mit zwei Studenten bewertet und verbessert. Um das Problem der zusätzlichen Laufwege zu lösen die entstehen wenn eine Person suchend von Raum zu Raum läuft, haben wir mit Hilfe mehrerer verketteter Listen und dem Dijkstra-Algorithmus aus der Graphentheorie einen Algorithmus konzipiert der den Raum mit der kürzesten Strecke zum Benutzer ermitteln kann. Um diese Wegbestimmung zu realisieren wird der Standort des Benutzer über Bluetooth Beacons ermittelt die in der gesamten Lehreinrichtung verteilt sind. Dazu scannt das Endgerät des Benutzers automatisch die Umgebung nach Bluetooth Signalen und liest die dazugehörige ID aus. Mit ihr weiß das System wo der Benutzer sich befindet, und kann so eine realistische Berechnung aufstellen. Um auch Gegenstände innerhalb von Räumen in der Raumauswahl zu berücksichtigen haben wir die Möglichkeit einer Markierung von Gegenständen durch RFID-Tags in Betracht gezogen. Dabei wird durch ein Lesegerät im Eingangsbereich eines jeden Raumes überprüft ob ein Gegenstand den Raum verlässt oder betritt. Dadurch wird eine dynamische Verschiebung von Equipment zwischen verschiedenen Räumen ermöglicht, was die Raumgestaltung flexibler gestaltet. Um zu überprüfen ob sich Personen innerhalb eines Raumes aufhalten womit der Status des Raumes festgelegt werden kann, haben wir uns für die Bildanalyse entschieden. Unserer Meinung nach wurden alle aufgestellten Ziele adressiert und deren Erfüllung im System durch Einsatz verschiedener Technologien und Prinzipien erläutert. Bei den von uns betrachteten Technologien ergaben sich viele interessante Möglichkeiten die aber teilweise im aktuellen Projektkontext nicht anwendbar waren. Sollten sich die ermittelten Probleme in näherer Zukunft beheben, wären auch Technologien wie Druckplatten im Boden als Personenerkennung eine brauchbare Methode. ?



1. - freien Raum Anfragen
- Raum reservieren
- Raum buchen
- Raumbuchung stornieren
- Raumbuchung verlängern

2. - Anfragen an den Server
(aus 1. entnommen)

3. - Anfragen von Daten die zur
Verarbeitung der Anfrage aus
1. benötigt werden

4. - Rufe benötigte
Daten aus dem
Datenspeicher ab

5. - Sende benötigte
Daten an den DB-
Server

6. - Sende zur
Verarbeitung der
Anfrage aus 1.
benötigte Daten an
den Server

7. - Sende
verarbeitete Daten
der Anfrage aus 1.
an das Endgerät
des Nutzers

8. - Präsentation der
angeforderten
Informationen aus
1.

Abbildung 3: Präskriptives Kommunikationsmodell

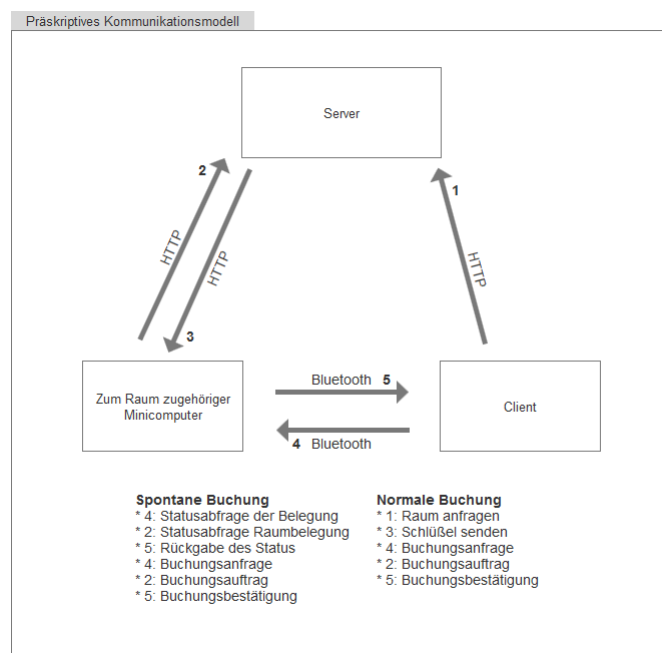


Abbildung 4: Präskriptives Architekturmodell

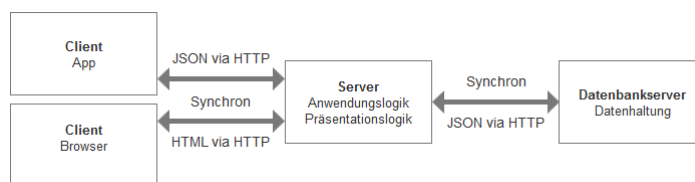


Abbildung 5: 3 Schichten Architektur

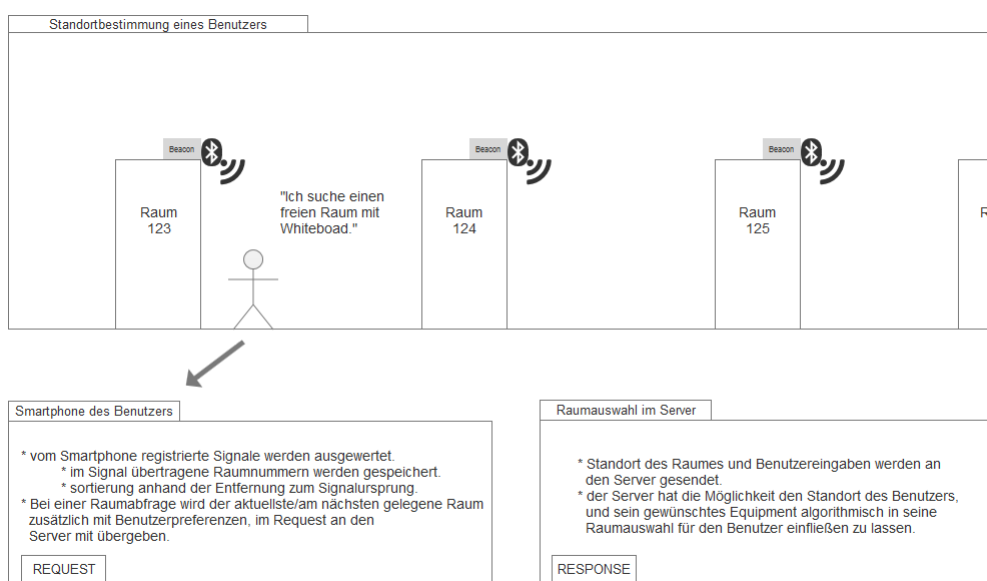


Abbildung 6: Standortbestimmung durch Beacons

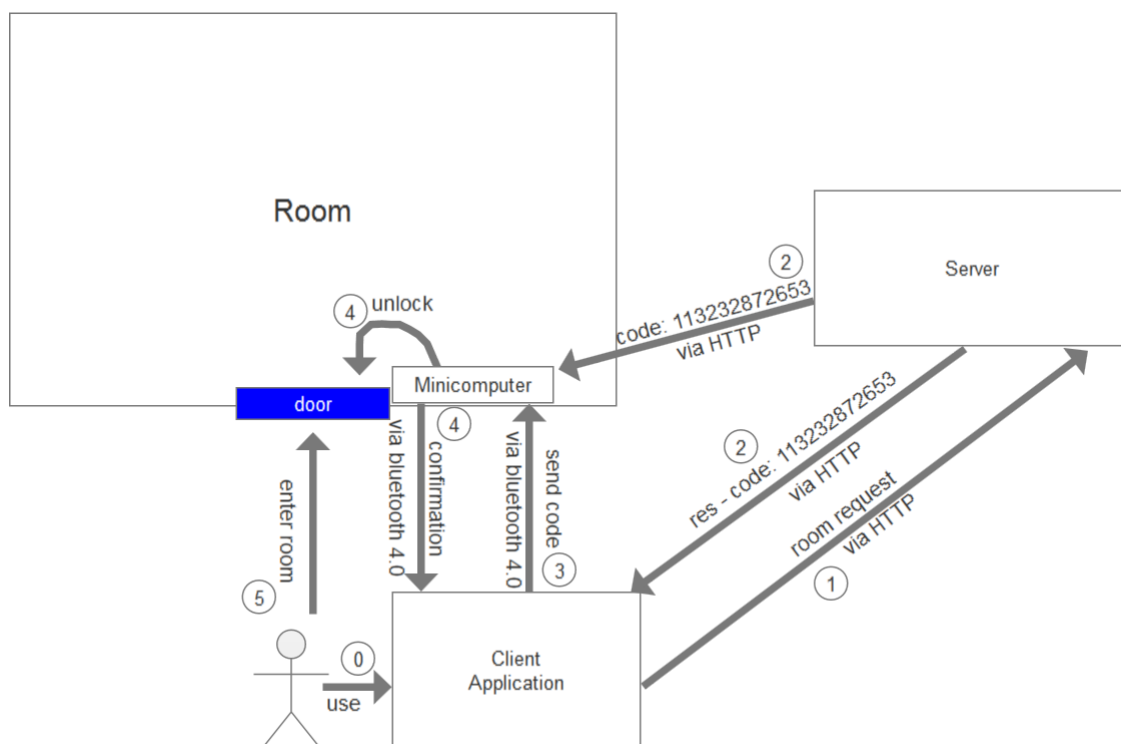


Abbildung 7: Zugang zum Raum

Literaturverzeichnis

- [COMTEC-Nöker GmbH 2017] COMTEC-NÖKER GMBH ; COMTEC-NÖKER GMBH (Hrsg.): *IN-TIME*. <https://www.comtec-noeker.de/>. Version: 2017
- [Constantine und Lockwood 1999] CONSTANTINE, Larry L. ; LOCKWOOD, Lucy A.: *Software For Use*. ACM Press and Addison-Wesley, 1999. – 582 S. – ISBN 0-201-92478-1
- [Estimote, Inc. 2017] ESTIMOTE, INC.: *Estimote, Inc.* https://estimote.com/?gclid=CjwKCAiAsejRBRB3EiwAZft7sJe4aSGlbgBgQP5SLAjT3GcxvFdEi754ZzeD8e9jIWHiyNdUdigPEBoC_5cQAvD_BwE. Version: Januar 2017
- [infsoft GmbH] INFISOFT GMBH: *infsoft GmbH - smart connected locations*. <https://www.infsoft.de/technologie/sensorik/bluetooth-low-energy-beacons>
- [LOY GmbH 2017] LOY GMBH ; LOY GMBH (Hrsg.): *Locaboo*. <https://www.locaboo.com/>. Version: 2017
- [OMOC.interactive 2017] OMOC.INTERACTIVE ; OMOC.INTERACTIVE (Hrsg.): *Online-Raumverwaltung*. <https://www.online-raumverwaltung.de/>. Version: 2017
- [TH Köln 2009] TH KÖLN ; TH KÖLN (Hrsg.): *TH Köln - Freie Lehrräume*. http://advbs06.gm.fh-koeln.de:7777/plsql/sport.raum_abfrage. Version: 2009

Eidesstattliche Erklärung

Wir, Bastian Fuchshofer und Niklas Fonseca Luis, versichern hiermit, dass wir unsere **Entwicklungsprojekt Interaktiver Systeme WS 17/18** mit dem Thema

Entwicklung eines Raumbuchungssystems – Webbasiertes Tool mit verteilter Anwendungslogik zur effektiven Raumnutzung

selbständig verfasst und keine anderen als die angegebenen Quellen mit Hilfsmitteln benutzt haben, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Gummersbach, dem 22.12.2017

BASTIAN FUCHSHOFER

NIKLAS FONSECA LUIS

A Anhang

A.1 Use Cases

A.2 Anwendungsfälle (Task Modelling) Version 1

- Benutzeraufgaben
 - Der Benutzer identifiziert sich im System.
 - * Identifikation des Benutzers durch ein Merkmal.
 - Der Benutzer sucht einen beliebigen freien Raum den er nutzen kann.
 - * Ein zufälliger Raum soll ausgegeben werden da keine besonderen Präferenzen vorliegen.
 - Der Benutzer sucht einen ruhigen Raum in dem er Arbeiten kann.
 - * Ein Raum zur stillen Einzelarbeit ist gesucht. Mehrere Einzelpersonen in einem Raum möglich.
 - Der Benutzer sucht einen Raum für mehrere Personen den er nutzen kann.
 - * Gruppenarbeitsraum für mehrere Personen zum arbeiten wird gesucht.
 - Der Benutzer sucht für mehrere Personen, mehrere freie Räume, die er nutzen kann.
 - * Anwendbar wenn z. B. ein Dozent für seine Studenten Räume zum arbeiten sucht.
 - Der Benutzer sucht einen freien Raum mit bestimmten Equipment den er nutzen kann.
 - * Verschiedene Arbeitsmittel sind gesucht die der Benutzer für seine arbeiten benötigt.
 - Der Benutzer sucht einen freien Raum der die vom Benutzer gewünschte Größe entspricht und den er nutzen kann.
 - * Raum für eine bestimmte Personenanzahl gesucht, bspw. bei Präsentationen.
- Administratöraufgaben
 - Der Administrator möchte einen Raum im System hinzufügen.
 - * Neue Anbauten oder bisher nicht relevante Räume sollen dem System hinzugefügt werden.
 - Der Administrator möchte einen Raum aus dem System entfernen.
 - * Ein Raum soll nicht mehr genutzt werden oder wurde umfunktioniert.
 - Der Administrator möchte einen bestehenden Raum im System mit neuen Informationen anreichern.
 - * Bspw. ein Raum bekommt einen Beamer installiert.
 - Der Administrator möchte Informationen über einen bestehenden Raum im System abändern.
 - * Ein Beamer in einem Raum ist defekt und muss aus der Inventarliste komplett entfernt werden.

A.3 Essential Use Cases Version 1

A.3.1 Administratöraufgaben

Tabelle 1: Essential Use Cases - Administratöraufgaben

Der Administrator möchte einen Raum im System hinzufügen - *addRoom*

User Intention	System Responsibility
Dem System mitteilen welcher Raum im System hinzugefügt werden soll	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Administrator möchte einen Raum aus dem System entfernen - *deleteRoom*

User Intention	System Responsibility
dem System mitteilen welcher Raum aus dem System entfernt werden soll	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Administrator möchte einen bestehenden Raum im System mit neuen Informationen anreichern - *addRoomInformations*

User Intention	System Responsibility
dem System mitteilen welcher Raum im System welche neuen Informationen erhalten soll	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Administrator möchte Informationen über einen bestehenden Raum im System abändern - *updateRoomInformations*

User Intention	System Responsibility
dem System mitteilen welche Informationen eines bestimmten Raumes im System geändert werden soll	gibt Statusmeldung über den Erfolg an den Benutzer zurück

A.3.2 Benutzeraufgaben

Tabelle 2: Essential Use Cases - Benutzeraufgaben

Der Benutzer identifiziert sich im System - *identify*

User Intention	System Responsibility
dem System mitteilen in welcher Beziehung er zu ihm steht	überprüft Berechtigungen des Benutzers

Der Benutzer sucht einen beliebigen freien Raum den er nutzen kann - *searchingStandardRoom*

User Intention	System Responsibility
dem System mitteilen das er einen beliebigen freien Raum sucht	Raumidentifikation an den Benutzer zurückgeben

Der Benutzer sucht einen ruhigen Raum in dem er Arbeiten kann - *searchingSilentRoom*

User Intention	System Responsibility
dem System mitteilen das er einen ruhigen freien Raum sucht	Raumidentifikation an den Benutzer zurückgeben

Der Benutzer sucht einen Raum für mehrere Personen den er nutzen kann - *searchingRoomForSeveralPersons*

User Intention	System Responsibility
dem System mitteilen das er einen Raum für mehrere Personen sucht	Raumidentifikation an den Benutzer zurückgeben

Der Benutzer sucht für mehrere Personen, mehrere freie Räume, die er nutzen kann - *searchingSeveralRoomsForSeveralPersons*

User Intention	System Responsibility
dem System mitteilen das er mehrere Räume für mehrere Personen sucht	Raumidentifikationen an den Benutzer zurückgeben

Der Benutzer sucht einen freien Raum mit bestimmten Equipment den er nutzen kann - *searchingRoomWithSpecificEquipment*

User Intention	System Responsibility
dem System mitteilen das er einen Raum mit vom Benutzer definierten Equipment sucht	Raumidentifikation an den Benutzer zurückgeben

Der Benutzer sucht einen freien Raum der die vom Benutzer gewünschte Größe entspricht und den er nutzen kann - *searchingRoomWithSpecificSize*

User Intention	System Responsibility
dem System mitteilen das er einen Raum mit vom Benutzer definierter Größe sucht	Raumidentifikation an den Benutzer zurückgeben

A.4 Narrative Concrete Use Cases Version 1**A.4.1 Administratöraufgaben**

Tabelle 3: Narrative Concrete Use Cases - Administratöraufgaben

Der Administrator möchte einen Raum im System hinzufügen - *addRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen neuen Raum im System hinzufügen möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen für diese Aktion, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und speichert die verarbeiteten Information innerhalb des persistenten Datenspeichers des Systems.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Administrator möchte einen Raum aus dem System entfernen - *deleteRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen im persistenten Datenspeicher vorhandenen Raum aus dem System entfernen möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Trifft das zu, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und speichert die verarbeiteten Information innerhalb des persistenten Datenspeichers des Systems.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Administrator möchte einen bestehenden Raum im System mit neuen Informationen anreichern
 - *addRoomInformations*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen im persistenten Datenspeicher vorhandenen Raum mit neuen Informationen anreichern möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Trifft das zu, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und speichert die verarbeiteten Information innerhalb des persistenten Datenspeichers des Systems.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Administrator möchte Informationen über einen bestehenden Raum im System abändern - *updateRoomInformations*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er Informationen zu einem Raum innerhalb des persistenten Datenspeichers des Systems ändern möchte . Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Trifft das zu, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und speichert die verarbeiteten Information innerhalb des persistenten Datenspeichers des Systems.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

A.4.2 Benutzeraufgaben

Tabelle 4: Narrative Concrete Use Cases - Benutzeraufgaben

Der Benutzer identifiziert sich im System - *identify*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit in welcher Beziehung er zu dem System steht. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	Das System identifiziert den Benutzer anhand bestimmter Merkmale und prüft somit welche Berechtigungen der Benutzer innerhalb des Systems besitzt.

Der Benutzer sucht einen beliebigen freien Raum den er nutzen kann - *searchingStandardRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen beliebigen freien Raum benötigt in dem er arbeiten kann. Dazu übermittelt der Benutzer automatisch dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

Der Benutzer sucht einen ruhigen Raum in dem er Arbeiten kann - *searchingSilentRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen ruhigen Raum benötigt in dem er arbeiten kann. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

Der Benutzer sucht einen Raum für mehrere Personen - *searchingRoomForSeveralPersons*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen Raum benötigt in dem er mit mehreren Personen arbeiten kann. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

Der Benutzer sucht für mehrere Personen, mehrere freie Räume, die er nutzen kann - *searchingSeveralRoomsForSeveralPersons*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er mehrere Räume benötigt in denen er mit mehreren Personen arbeiten kann. Dazu übermittelt er dem System die dafür benötigten Informationen	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummern von Räumen die den Wunsch des Benutzers entsprechen.
	Im Anschluss präsentiert das System dem Benutzer die identifizierten Raumnummern.

Der Benutzer sucht einen freien Raum mit bestimmten Equipment den er nutzen kann - *searchingRoomWithSpecificEquipment*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen Raum benötigt in dem bestimmtes Equipment vorhanden ist das er zum arbeiten benötigt. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

Der Benutzer sucht einen freien Raum der die vom Benutzer gewünschte Größe entspricht und den er nutzen kann - *searchingRoomWithSpecificSize*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen Raum benötigt der eine bestimmte Größe besitzt um darin arbeiten zu können. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

A.5 Anwendungsfälle (Task Modelling) Version 2

- Benutzeraufgaben
 - Der Benutzer identifiziert sich im System.
 - * Identifikation des Benutzers durch ein Merkmal.
 - Der Benutzer sucht einen beliebigen freien Raum den er nutzen kann
 - * Ein zufälliger Raum soll ausgegeben werden da keine besonderen Präferenzen vorliegen
 - Der Benutzer sucht einen ruhigen Raum in dem er Arbeiten kann.
 - * Ein Raum zur stillen Einzelarbeit ist gesucht. Mehrere Einzelpersonen in einem Raum möglich.
 - Der Benutzer sucht einen Raum für mehrere Personen den er nutzen kann.
 - * Gruppenarbeitsraum für mehrere Personen zum arbeiten wird gesucht.
 - Der Benutzer sucht für mehrere Personen, mehrere freie Räume, die er nutzen kann.
 - * Anwendbar wenn z.B. ein Dozent für seine Studenten Räume zum arbeiten sucht.
 - Der Benutzer sucht einen freien Raum mit bestimmten Equipment den er nutzen kann.
 - * Verschiedene Arbeitsmittel sind gesucht die der Benutzer für seine arbeiten benötigt.
 - Der Benutzer sucht einen freien Raum der die vom Benutzer gewünschte Größe entspricht und den er nutzen kann.
 - * Raum für eine bestimmte Personenanzahl gesucht, bspw. bei Präsentationen.
 - Der Benutzer möchte den Status eines bestimmten Raumes erfahren.
 - * Information ob ein Raum gerade frei, reserviert oder gebucht ist. Relevant bei bspw. CleaningStaff/MaintainStaff
 - Der Benutzer möchte einen freien Raum für einen bestimmten Zeitraum für andere Benutzer sperren.
 - * Dieser Raum ist für andere Personen als nicht verfügbar gekennzeichnet.
 - Der Benutzer möchte einen freien Raum für eine vorgegebene Zeitspanne reservieren.
 - * Um zu verhindern das der vom System gefundene Raum während der Benutzer zu ihm geht von anderen Personen belegt wird.
 - Der Benutzer möchte einen freien Raum für eine vorgegebene Zeitspanne buchen.
 - * Ist ein Raum gebucht kann er vom Benutzer für seine arbeiten verwendet werden.
 - Der Benutzer möchte einen von ihm gesperrten Raum wieder freigeben.
 - * Folgeaktion vom sperren eines Raumes.
 - Der Benutzer möchte einen von ihm gefundenen freien Raum buchen.
 - * Das Buchen eines Raumes ermöglichen der nicht vom System vorgeschlagen sondern vom Benutzer ausgewählt wurde, bspw. durch Vorabbuchung in der Zukunft.
 - Der Benutzer möchte die Reservierung für einen von ihm reservierten Raum stornieren.
 - * Vom System ausgegebener Raum wird spontan doch nicht mehr benötigt und soll wieder für andere Benutzer freigegeben werden.

- Der Benutzer möchte die Buchung für einen von ihm gebuchten Raum stornieren.
 - * Arbeiten werden vorzeitig beendet und der Raum wird nicht länger benötigt.
- Der Benutzer möchte die Buchung für einen von ihm gebuchten Raum verlängern.
 - * Die arbeiten des Benutzers dauern länger als erwartet und der Raum zwecks Bequemlichkeit weiter genutzt werden anstat einen neuen Raum zu suchen.
- Administratöraufgaben
 - Der Administrator möchte einen Raum im System hinzufügen.
 - * Neue Anbauten oder bisher nicht relevante Räume sollen dem System hinzugefügt werden.
 - Der Administrator möchte einen Raum aus dem System entfernen.
 - * Ein Raum soll nicht mehr genutzt werden oder wurde umfunktioniert.
 - Der Administrator möchte einen bestehenden Raum im System mit neuen Informationen anreichern.
 - * Bspw. ein Raum bekommt einen Beamer installiert.
 - Der Administrator möchte Informationen über einen bestehenden Raum im System abändern.
 - * Ein Beamer in einem Raum ist defekt und muss aus der Inventarliste komplett entfernt werden.

A.6 Essential Use Cases Version 2

A.6.1 Administratöraufgaben

Tabelle 5: Essential Use Cases - Administratöraufgaben

Der Administrator möchte einen Raum im System hinzufügen - *addRoom*

User Intention	System Responsibility
Dem System mitteilen welcher Raum im System hinzugefügt werden soll	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Administrator möchte einen Raum aus dem System entfernen - *deleteRoom*

User Intention	System Responsibility
dem System mitteilen welcher Raum aus dem System entfernt werden soll	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Administrator möchte einen bestehenden Raum im System mit neuen Informationen anreichern - *addRoomInformations*

User Intention	System Responsibility
dem System mitteilen welcher Raum im System welche neuen Informationen erhalten soll	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Administrator möchte Informationen über einen bestehenden Raum im System abändern - *updateRoomInformations*

User Intention	System Responsibility
dem System mitteilen welche Informationen eines bestimmten Raumes im System geändert werden soll	gibt Statusmeldung über den Erfolg an den Benutzer zurück

A.6.2 Benutzeraufgaben

Tabelle 6: Essential Use Cases - Benutzeraufgaben

Der Benutzer identifiziert sich im System - *identify*

User Intention	System Responsibility
dem System mitteilen in welcher Beziehung er zu ihm steht	überprüft Berechtigungen des Benutzers

Der Benutzer sucht einen beliebigen freien Raum den er nutzen kann - *searchingStandardRoom*

User Intention	System Responsibility
dem System mitteilen das er einen beliebigen freien Raum sucht	Raumidentifikation an den Benutzer zurückgeben

Der Benutzer sucht einen ruhigen Raum in dem er Arbeiten kann - *searchingSilentRoom*

User Intention	System Responsibility
dem System mitteilen das er einen ruhigen freien Raum sucht	Raumidentifikation an den Benutzer zurückgeben

Der Benutzer sucht einen Raum für mehrere Personen den er nutzen kann - *searchingRoomForSeveralPersons*

User Intention	System Responsibility
dem System mitteilen das er einen Raum für mehrere Personen sucht	Raumidentifikation an den Benutzer zurückgeben

Der Benutzer sucht für mehrere Personen, mehrere freie Räume, die er nutzen kann - *searchingSeveralRoomsForSeveralPersons*

User Intention	System Responsibility
dem System mitteilen das er mehrere Räume für mehrere Personen sucht	Raumidentifikationen an den Benutzer zurückgeben

Der Benutzer sucht einen freien Raum mit bestimmten Equipment den er nutzen kann - *searchingRoomWithSpecificEquipment*

User Intention	System Responsibility
dem System mitteilen das er einen Raum mit vom Benutzer definierten Equipment sucht	Raumidentifikation an den Benutzer zurückgeben

Der Benutzer sucht einen freien Raum der die vom Benutzer gewünschte Größe entspricht und den er nutzen kann - *searchingRoomWithSpecificSize*

User Intention	System Responsibility
dem System mitteilen das er einen Raum mit vom Benutzer definierter Größe sucht	Raumidentifikation an den Benutzer zurückgeben

Tabelle 7: Essential Use Cases - Benutzeraufgaben

Ergänzung der Benutzeraufgaben anhand der Benutzerevaluation

Der Benutzer möchte wissen ob ein bestimmter Raum gerade reserviert, gebucht oder gesperrt ist - *checkRoomStatus*

User Intention	System Responsibility
dem System mitteilen das er den Status eines bestimmten Raumes wissen möchte	gibt Status des Raumes an den Benutzer zurück

Der Benutzer möchte einen von ihm gefundenen freien Raum buchen - *bookSpecificRoom*

User Intention	System Responsibility
dem System mitteilen das er einen bestimmten Raum buchen möchte	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Benutzer möchte einen freien Raum für einen bestimmten Zeitraum für andere Benutzer blockieren - *lockRoom*

User Intention	System Responsibility
dem System mitteilen das er einen bestimmten Raum für andere Benutzer blockieren möchte	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Benutzer möchte einen, vom System vorgeschlagenen freien Raum für eine vorgegebene Zeitspanne reservieren - *reservateGivenRoom*

User Intention	System Responsibility
dem System mitteilen das er einen vom System vorgegebenen Raum reservieren möchte	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Benutzer möchte einen, vom System vorgeschlagenen freien Raum für eine vorgegebene Zeitspanne buchen - *bookGivenRoom*

User Intention	System Responsibility
dem System mitteilen das er einen vom System vorgegebenen Raum buchen möchte	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Benutzer möchte einen von ihm gesperrten Raum wieder freigeben - *unlockRoom*

User Intention	System Responsibility
dem System mitteilen das er einen bestimmten Raum für andere Benutzer wieder freigeben möchte	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Benutzer möchte die Reservierung für einen von ihm reservierten Raum stornieren - *cancelRoomReservation*

User Intention	System Responsibility
dem System mitteilen das er einen von ihm reservierten Raum stornieren möchte	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Tabelle 8: Essential Use Cases - Benutzeraufgaben

Der Benutzer möchte die Buchung für einen von ihm gebuchten Raum stornieren - *cancelRoomBooking*

User Intention	System Responsibility
dem System mitteilen das er einen von ihm gebuchten Raum stornieren möchte	gibt Statusmeldung über den Erfolg an den Benutzer zurück

Der Benutzer möchte die Buchung für einen von ihm gebuchten Raum verlängern - *extendRoomBooking*

User Intention	System Responsibility
dem System mitteilen das er einen von ihm reservierten Raum stornieren möchte	gibt Statusmeldung über den Erfolg an den Benutzer zurück

A.7 Narrative Concrete Use Cases Version 2

A.7.1 Administratöraufgaben

Tabelle 9: Narrative Concrete Use Cases - Administratöraufgaben

Der Administrator möchte einen Raum im System hinzufügen - *addRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen neuen Raum im System hinzufügen möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen für diese Aktion, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und speichert die verarbeiteten Information innerhalb des persistenten Datenspeichers des Systems.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Administrator möchte einen Raum aus dem System entfernen - *deleteRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen im persistenten Datenspeicher vorhandenen Raum aus dem System entfernen möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Trifft das zu, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und speichert die verarbeiteten Information innerhalb des persistenten Datenspeichers des Systems.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Administrator möchte einen bestehenden Raum im System mit neuen Informationen anreichern
 - *addRoomInformations*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen im persistenten Datenspeicher vorhandenen Raum mit neuen Informationen anreichern möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Trifft das zu, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und speichert die verarbeiteten Information innerhalb des persistenten Datenspeichers des Systems.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Administrator möchte Informationen über einen bestehenden Raum im System abändern - *updateRoomInformations*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er Informationen zu einem Raum innerhalb des persistenten Datenspeichers des Systems ändern möchte . Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Trifft das zu, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und speichert die verarbeiteten Information innerhalb des persistenten Datenspeichers des Systems.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

A.7.2 Benutzeraufgaben

Tabelle 10: Narrative Concrete Use Cases - Benutzeraufgaben

Der Benutzer identifiziert sich im System - *identify*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit in welcher Beziehung er zu dem System steht. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	Das System identifiziert den Benutzer anhand bestimmter Merkmale und prüft somit welche Berechtigungen der Benutzer innerhalb des Systems besitzt.

Der Benutzer sucht einen beliebigen freien Raum den er nutzen kann - *searchingStandardRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen beliebigen freien Raum benötigt in dem er arbeiten kann. Dazu übermittelt der Benutzer automatisch dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

A Anhang

Der Benutzer sucht einen ruhigen Raum in dem er Arbeiten kann - *searchingSilentRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen ruhigen Raum benötigt in dem er arbeiten kann. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

Der Benutzer sucht einen Raum für mehrere Personen - *searchingRoomForSeveralPersons*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen Raum benötigt in dem er mit mehreren Personen arbeiten kann. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

Ergänzung der Benutzeraufgaben anhand der Benutzerevaluation

Der Benutzer sucht für mehrere Personen, mehrere freie Räume, die er nutzen kann - *searchingSeveralRoomsForSeveralPersons*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er mehrere Räume benötigt in denen er mit mehreren Personen arbeiten kann. Dazu übermittelt er dem System die dafür benötigten Informationen	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummern von Räumen die den Wunsch des Benutzers entsprechen.
	Im Anschluss präsentiert das System dem Benutzer die identifizierten Raumnummern.

Der Benutzer sucht einen freien Raum mit bestimmten Equipment den er nutzen kann - *searchingRoomWithSpecificEquipment*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen Raum benötigt in dem bestimmtes Equipment vorhanden ist das er zum arbeiten benötigt. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

Der Benutzer sucht einen freien Raum der die vom Benutzer gewünschte Größe entspricht und den er nutzen kann - *searchingRoomWithSpecificSize*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen Raum benötigt der eine bestimmte Größe besitzt um darin arbeiten zu können. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

Der Benutzer möchte wissen ob ein bestimmter Raum reserviert, gebucht oder gesperrt ist - *checkRoomStatus*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er den Status eines Raumes erfahren möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt den Status des Raumes.
	Im Anschluss präsentiert das System dem Benutzer den Status des Raumes.

Der Benutzer möchte einen von ihm gefundenen freien Raum buchen - *bookSpecificRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen vom Benutzer gesperrten Raum wieder freigeben möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und gibt den Raum wieder frei.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Benutzer möchte einen freien Raum für einen bestimmten Zeitraum für andere Benutzer sperren
 - *lockRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen bestimmten Raum für einen bestimmten Zeitraum für andere Benutzer sperren möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und ermittelt die Raumnummer eines Raumes der den Wunsch des Benutzers entspricht.
	Im Anschluss präsentiert das System dem Benutzer die identifizierte Raumnummer.

Der Benutzer möchte einen, vom System vorgeschlagenen freien Raum für eine vorgegebene Zeitspanne reservieren - *reservateGivenRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen vom System vorgeschlagenen bestimmten Raum reservieren möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und reserviert den Raum. Die Reservierung wird im persistenten Speicher des Systems hinterlegt.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Benutzer möchte einen, vom System vorgeschlagenen freien Raum für eine vorgegebene Zeitspanne buchen - *bookGivenRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen vom System vorgeschlagenen bestimmten Raum buchen möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und reserviert den Raum.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg der Aktion.

Der Benutzer möchte einen von ihm gesperrten Raum wieder freigeben - *unlockRoom*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen vom Benutzer gesperrten Raum wieder freigeben möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und gibt den Raum wieder frei.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Tabelle 11: Der Benutzer möchte die Reservierung für einen von ihm reservierten Raum stornieren - *cancelRoomReservation*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen vom Benutzer reservierten Raum stornieren möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und storniert die Raumreservierung.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Benutzer möchte die Buchung für einen von ihm gebuchten Raum stornieren - *cancelRoomBooking*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen vom Benutzer gebuchten Raum stornieren möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und storniert die Raumbuchung.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

Der Benutzer möchte die Buchung für einen von ihm gebuchten Raum verlängern - *extendRoomBooking*

User Intention	System Responsibility
Der Benutzer interagiert mit dem System und teilt ihm mit das er einen vom Benutzer gebuchten Raum verlängern möchte. Dazu übermittelt er dem System die dafür benötigten Informationen.	
	>identify
	Besitzt der Benutzer die benötigten Berechtigungen, wertet das System die vom Benutzer spezifizierten Informationen aus, verarbeitet diese und leitet die verlängerung der Raumbuchung ein.
	Im Anschluss präsentiert das System dem Benutzer eine Statusmeldung über den Erfolg oder Misserfolg der Aktion.

A.8 User Roles

- **FreeRoomSearcher**

- häufige Verwendung
- einfache Nutzung
- wenige interaktionsschritte

- * **SilentWorkingRoomSearcher**

- sucht für sofort einen freien Raum
 - sucht stille Arbeitsmöglichkeit

- * **CasualSingleRoomSearcher**

- sucht für sofort einen freien Raum
 - sucht Arbeitsmöglichkeiten für mehrere Personen

- * **CasualMultiRoomSearcher**

- sucht für sofort mehrere freie Räume
 - sucht Arbeitsmöglichkeiten für mehrere Personen

- * **CasualRoomWithEquipmentSearcher**

- sucht für sofort einen freien Raum mit gewünschten Rauminhalten
 - sucht Arbeitsmöglichkeiten

- * **RoomWithProfessionalEquipmentSearcher**

- sucht für sofort einen freien Raum mit gewünschten speziellen Rauminhalten

- * **SpecificRoomBooker**

- direktes buchen eines Raumes

- * **RoomScheduler**

- sucht für einen bestimmten Zeitraum in der Zukunft einen freien Raum
 - verantwortungsbewusste Raumreservierung

- **Administrator**

- einfache Nutzung
- effiziente Ausführung der Arbeitsaufgabe

- Fehleranalyse
- hohe Toleranz bezüglich häufig verwendeter Formate
- * **DBAdministrator**
 - Verwaltung und Aktualisierung von Informationen innerhalb eines persistenten Datenspeichers
- * **ServerAdministrator**
 - Verwaltung und Wartung des/der Server(s) des Systems
- **RoomStatusChecker**
 - einfache Nutzung
 - häufige Verwendung
 - wenige Interaktionsschritte
 - den Status eines bestimmten Raumes abfragen
- * **SpecificRoomBooker**
 - direktes buchen eines Raumes
- * **RoomScheduler**
 - sucht für einen bestimmten Zeitraum in der Zukunft einen freien Raum
 - verantwortungsbewusste Raumreservierung
- * **SpecificRoomBlocker**
 - temporäres blockieren von Räumen zwecks Wartungs-/Reinigungsarbeiten
 - **CleaningStaff**
 - **MaintainStaff**

A.8.1 Focal Roles

Focal roles are those few user roles judged to be the most common or typical or that are deemed particularly important from some other perspective. [CONSTANTINE UND LOCKWOOD, 1999, Seite 79]

- CasualSingleRoomSearcher

A.9 Content List

- **userSpace**
 - **Start Dialogue**
 - * *toolSelection*
Eine Auswahl an verfügbaren Funktionen und Tools.
 - **Filter Dialogue**
 - * *roomEquipment*
Eine Auswahl an möglichem Equipment.
 - * *roomProperties*
Eine Auswahl an möglichen Raumeigenschaften.
 - * *roomSize*
Auswahl der Größe des Raumes.
 - **Reservation Dialogue**
 - * *roomNumber*
Anzeige der Identifikation des reservierten Raumes.
 - * *reservationDuration*
Anzeige der restlichen Dauer der Reservierung.
 - **Multi Reservation Dialogue**
 - * *roomNumberList*
Anzeige der Identifikationen der reservierten Räume.
 - * *reservationDuration*
Anzeige der restlichen Dauer der Reservierungen.
 - **Booking Dialogue**
 - * *roomNumber*
Anzeige der Identifikation des gebuchten Raumes.
 - * *bookingDuration*
Anzeige der restlichen Dauer der Buchung.
 - **Multi Booking Dialogue**
 - * *roomNumberList*
Anzeige der Identifikationen der gebuchten Räume.
 - * *bookingDuration*
Anzeige der restlichen Dauer der Reservierungen.
- **instituteSpace**
 - **Start Dialogue**
 - * *toolSelection*
Eine Auswahl an verfügbaren Funktionen und Tools.
 - **Search Dialogue**

- **Block Dialogue**
 - * *roomNumber*
Anzeige der Identifikation des geblockten Raumes.
 - * *blockingDuration*
Anzeige der restlichen Dauer der Blockung.
- **adminSpace**
 - **Start Dialogue**
 - * *toolSelection*
Eine Auswahl der nur für den Administrator verfügbaren Funktionen und Tools.
 - **Search Dialogue**
 - **Add Room Dialogue**
 - * *roomInformationForm*
Eine Formular mit notwendigen und optionalen Raumeigenschaften und Rauminformationen.
 - **Edit Room Dialogue**
 - * *roomInformationForm*
Ein Formular mit allen im System verfügbaren Rauminformationen.
 - **Delete Room Dialogue**
 - * *roomNumber*
Anzeige der Identifikation des zu löschenden Raumes
 - **Add Equipment Dialogue**
 - * *equipInformationForm*
Ein Formular mit notwendigen und optionalen Equipmenteigenschaften und Equipmentinformationen.
 - **Edit Equipment Dialogue**
 - * *equipInformationForm*
Ein Formular mit allen im System verfügbaren Equipmentinformationen.
 - **Delete Equipment Dialogue**
 - * *equipID*
Anzeige der Identifikation des zu löschenden Equipments.
 - **Add Eventplan Dialogue**
 - * *eventplanInformationForm*
Ein Formular mit notwendigen und optionalen Eventplaninformationen.
 - **Edit Eventplan Dialogue**
 - * *eventplanInformationForm*
Ein Formular mit allen im System verfügbaren Eventplaninformationen.
 - **Delete Eventplan Dialogue**

- * *eventplanInformation*
Anzeige der Identifikation des zu löschenden Eventplans.
- **Add Knoten Dialogue**
 - * *knotenInformationForm*
Ein Formular mit notwendigen und optionalen Knoteninformationen.
- **Edit Knoten Dialogue**
 - * *knotenInformationForm*
Ein Formular mit allen im System verfügbaren Knoteninformationen.
- **Delete Knoten Dialogue**
 - * *nodeID*
Anzeige der Identifikation des zu löschenden Knoten.

Erklärung der Dialoge

Start Dialogue Dieser Dialog soll für alle interaction spaces den Kontext beschreiben den der Benutzer aufruft wenn er auf das System zugreift. Von ihm aus kann auf die wichtigen Funktionen des Systems zugegriffen werden. Für den instituteSpace und userSpace wurde dieser Dialog zusammen gelegt, da diese einen ähnlichen Anwendungsbereich beinhalten.

Search Dialogue Der Search Dialogue des instituteSpaces stellt den Kontext der Suche nach einem bestimmtem Raum dar, d.h. die Suche mit Hilfe einer bekannten Raumnummer oder Raumeigenschaften. Der Search Dialogue des adminSpaces stellt den gleichen Kontext dar, erweitert um die Veranstaltungen, das Equipment und die Knotenpunkte mit der Bedingung dass die Identifikation dieser bekannt ist.

Reservation Dialogue / Multi Reservation Dialogue Diese Dialoge stellen den Kontext dar in dem sich ein Benutzer befindet, wenn er einen Raum reserviert hat, also dessen Raumnummer erhalten hat und sich auf dem Weg zu diesem befindet. Die Variante mit Multi ist lediglich der Kontext in dem sich ein Benutzer befindet wenn er mehrere Räume gleichzeitig reserviert hat, sofern er die benötigte Berechtigung besitzt.

Booking Dialogue / Multi Booking Dialogue Ähnlich wie die o.g. Reservierungs Dialoge beziehen sich diese auf den Kontext nach dem ein Nutzer einen Raum gebucht bzw. belegt hat. Die Variante mit Multi ist lediglich erneut der Kontext mit mehreren gebuchten bzw. belegten Räumen.

Add * / Edit * / Delete * Diese Dialoge umfassen die Kontexte die durch den Admin hervorgerufen werden um eine Ressource aus dem System zu verändern , hinzuzufügen oder zu entfernen.

A.10 Personenerkennung im Raum

A.11 Druckplatten im Boden

Bei der Zählung von Personen durch eine druckempfindliche Matte im Eingangs- Bereich kann durch Algorithmen sowohl die Anzahl der Personen als auch die Laufrichtung dieser durch Analyse der Drucksensoren bestimmt werden. Diese Matten können im Eingangsbereich zu den Räumen unter Fußmatten versteckt werden und sind robust genug so dass keine Schäden durch Equipment welches über diese gerollt wird entstehen sollte. Allerdings ist auf der Webseite des Anbieters [?] auf den wir uns bei dieser Methode beziehen keiner Information dazu zu finden ob ein Gegenstand, zum Beispiel ein Whiteboard mit rollen, beim Schieben über die Matte als Person erkannt wird oder nicht. Hierfür müsste sofern möglich ein eigenes System entwickelt oder das bestehende um Algorithmen zur Erkennung von bestimmten Merkmalen, wie z.B. des Reifenabstandes, und der Zuordnung dieser erweitert werden. Weiterhin kann es durch das Verrutschen der Matte dazu kommen das Türen nicht mehr richtig schließen. Dies kann durch das Befestigen der Matte auf dem Boden behoben werden. Außerdem muss die Matte eine zu ermittelnde Mindestgröße besitzen, damit Menschen mit größerer Schrittweite auch von dieser erkannt werden.

A.12 Lichtschranken und Bewegungsmelder

Einige Methoden können zu einer Gruppe zusammengeschlossen werden, da diese in etwa ähnlich funktionieren oder ähnliche Vor- und Nachteile besitzen. So bilden die Bewegungsmelder und Lichtschranken zum Beispiel eine solche Gruppe. Die Vorteile dieser Methoden sind zum Einen die Datenschutz unbedenkliche Verwendung, da durch diese Methoden weder Bild- noch Tonmaterial aufgenommen wird und somit auch keine Personenbezogenen Daten erhoben werden können. Der Große Nachteil dieser Methoden ist, dass durch die Verwendung eines einzelnen Gerätes weder die Laufrichtung noch die genaue Anzahl der Personen bestimmen kann. So kann es zum Beispiel dazu führen, dass mehrere Personen parallel oder leicht versetzt einen Raum betreten, vom System aber nur als eine Person erkannt werden, da die Sensoren nur einen durchgängigen Impuls senden. Des weiteren ist der große Abdeckungsbereich des Bewegungsmelders eher hinderlich, da es dadurch auch zu Fehldeutungen kommen kann, wenn ein Benutzer sich der Tür nur nähert, ohne hindurchgehen zu wollen. Dieses Problem könnte für den Bewegungsmelder gelöst werden, indem künstlich sein Wirkungsbereich eingedämmt wird. Außerdem könnte durch die Verwendung mehrerer Geräte sowohl eine genauere Bestimmung der Anzahl sowie der Laufrichtung von Personen bestimmt werden. Dies trifft nicht zu, wenn mehrere Personen gleichzeitig mit Unterschiedlichen Laufrichtungen hindurch laufen. Ein weiteres Problem tritt auf, wenn Gegenstände durch den Wirkungsbereich geschoben werden, da der Sensor nicht zwischen Mensch und Gegenstand unterscheiden kann. Um das Problem zu beseitigen könnte als die Dauer des Impulses gemessen werden und anhand dessen entscheidet das System ob es sich um einen Gegenstand oder eine Person handelt.

A.13 Zählmechanik

Die Methode eines Zählers oder einer Mechanik im Eingangsbereich oder der Tür haben wir als einzelne Methode ausgeschlossen, da zwar eine effektive Erkennung einer Interaktion mit einem Raum erkannt werden kann, jedoch nicht festgestellt werden kann, ob der Raum betreten oder verlassen wird, oder ob nur die Tür geöffnet wird. Es könnte durch die Kombination mit einer anderen eine effektivere Methode generiert werden. Weiterhin könnte durch Verwendung von Algorithmen bestimmt werden, ob eine Tür zum Beispiel mehrfach aufgehalten wird und anhand der Zeit die eine Tür offen ist, wie viele Personen möglicherweise hindurchgegangen sind. Dies führt allerdings zu keiner genauen Anzahl an Personen im Raum.

A.14 Wärmebildkameras

Eine effektive Methode welche auch die persönlichen Daten der Benutzer schützen würde wäre die Verwendung einer Wärmebildkamera mit zugehörigem Algorithmus zur Personenzählung. Diese erreicht abhängig vom Algorithmus eine sehr hohe Genauigkeit und ermöglicht wie zuvor gesagt die anonyme Zählung der Personen in einem Raum. Wir haben diese Methode allerdings in diesem Projekt ausgeschlossen, da durch die hohen Anschaffungskosten eine Verwendung im Nutzungskontext einer Lehreinrichtung mit mehreren hundert Räumen zum momentanen Zeitpunkt nicht denkbar ist. Durch die Weiterentwicklung dieser Technologie sollte die Verwendung, falls die Anschaffungskosten wesentlich geringer ausfallen, noch einmal im Rahmen dieses Projektes evaluiert werden.

A.15 Video-/Bildanalyse

Eine ähnliche und kostengünstigere Alternative bietet eine Analyse von Video/- bzw. Bildmaterial mit der durch algorithmische Personenerkennung die Anzahl der Personen ermittelt werden kann. Der wichtigste Unterschied zur Wärmebildkamera ist hierbei allerdings die fehlende Anonymität der Benutzer, da ein Video aus dem inneren des Raumes aufgenommen wird. Es kann ebenfalls durch Verwendung der richtigen Algorithmen eine hohe Genauigkeit erzielt, und so eine effektive Personenzählung durchgeführt werden. Da die Analyse solcher Videos sehr viel Rechenleistung benötigt und bei größeren Räumen die Videos ebenfalls hochauflösender sein müssen könnte in diesem Projekt eine Analyse von einzelnen Bildern, welche in festgelegten Intervallen aufgenommen werden, durchgeführt werden. Dadurch könnte eine gute Schätzung der Personen die sich in einem Raum befinden erreicht werden. Die Frage des Datenschutzes müsste bei der Umsetzung des Projektes mit dem Datenschutzbeauftragten der jeweiligen Lehreinrichtung besprochen werden. Allerdings sollte darauf geachtet werden dass die Bilder nur Lokal analysiert werden, danach gelöscht und nur das Ergebnis also zum Beispiel '2 Personen' weiterverwendet wird.

A.16 NFC

Eine weitere Möglichkeit der Benutzererkennung innerhalb von Räumen wären der Einsatz von NFC (Near Field Communication). Fast jedes neue Smartphone besitzt die Möglichkeit NFC einzusetzen. Bekannt ist diese Methode z.B. durch das kontaktlose Bezahlen mit einem Smartphone. Die Reichweite von NFC liegt aktuell nur bei wenigen Zentimetern ?, weswegen ein passives Erkennen von Benutzern nur schwer möglich ist. Der Benutzer müsste, um über NFC erkannt zu werden, sein Endgerät sehr nahe an das Lesegerät halten, was eine aktive Interaktion des Benutzers voraussetzt. In unserem Projektkontext halten wir diese Methode für kontraproduktiv, da der Benutzer sich aktiv am Scanner verifizieren muss was eine zusätzliche Aktivität bedeutet.

A.17 BLE Beacons

Der Einsatz von Beacons wurde bereits bei der Standortbestimmung von Benutzern angesprochen. Ebenfalls anwendbar wäre diese Methode bei der Bestimmung von Benutzern innerhalb von Räumen. Es wird also nach dem Standort innerhalb eines bestimmten Bereiches gesucht. Dabei müsste das Endgerät des Benutzers als sendender Beacon umfunktioniert werden und ein Empfangsgerät scannt innerhalb eines Raumes nach Bluetooth Signalen. Sofern sich innerhalb des Raumes bzw. in einem bestimmten Umkreis des Empfängers Bluetooth Signale empfangen lassen, kann das System davon ausgehen das sich eine oder mehrere Personen im Raum befinden. Sobald das Signal des Benutzers eine bestimmte Entfernung überschreitet, weiß das System das dieser Benutzer den Raum vermutlich verlassen hat. Falls das Signal des Benutzers wegen Störungen im Frequenzbereich vorübergehend nicht erreicht werden kann, sollte das System den Benutzer nicht sofort als nicht mehr vorhanden , bzw. den Raum als leer kennzeichnen, sondern eine gewisse Zeitspanne einräumen in der weiter nach Benutzersignalen innerhalb eines Raumes gescannt wird. Ist nach Ablauf dieser Zeit kein Benutzer erkannt worden, wird der Raum wieder im System für andere Benutzer freigegeben. Der offensichtliche Nachteil dieser Technologie für die Benutzererkennung ist, dass nicht jeder Benutzer zwingend ein empfangsfähiges Endgerät dabei haben muss. Da immer nur ein Benutzer einen Raum für z.B. eine Gruppe von Personen bucht, besteht die Gefahr das die Anzahl der Personen im Raum nicht erkannt werden kann. Verlässt der Benutzer der den Raum gebucht hat den Raum wird er vom System nicht mehr erkannt und der Raum wird wieder freigegeben obwohl sich noch arbeitende Personen im Raum aufhalten. Ein zusätzliches Problem besteht im gelegentlichen verlassen des Raumes, z.B. für den Toilettenbesuch. Wird die im System hinterlegte Zeit überschritten, wird der Raum wieder freigegeben obwohl das vom Benutzer eigentlich nicht gewollt ist.

A.18 Ressourcen

Tabelle 12: Benutzer

Ressource	Methode	Semantik	content-type (req)	content-type (res)
/freeRoom	GET	gibt einen Raum zurück der in Abhängigkeit der Informationen im Body ausgewählt wurde.	application/json	application/json
/room/{roomId}	GET	gibt den aktuellen Status eines Raumes aus	-	application/json
/room/{roomId}/status	PUT	ändert den Status des angegebenen Raumes in den angegebenen Status	application/json	application/json

Tabelle 13: Administrator

Ressource	Methode	Semantik	content-type (req)	content-type (res)
/room	PUT	gibt einen Raum zurück der in Abhängigkeit der Informationen im Body ausgewählt wurde.	application/json	application/json
/room/{roomID}	PUT	gibt den aktuellen Status eines Raumes aus	-	application/json
/room/{roomID}	DELETE	entfernt einen Raum aus dem System	-	-
/room/{roomID}	GET	enthält alle Informationen die zu einem Raum gespeichert wurden	-	application/json

A.19 Stakeholderanalyse

Tabelle 14: Stakeholderanalyse

Bezeichnung	Beziehung zum System	Objektbereich	Erfordernis, Erwartung	Prio.
Lehrkraft	Anspruch	Rauminformation	Als Lehrkraft muss ich die korrekten Rauminformationen verfügbar haben um einen Raum als für meine Arbeitsaufgabe passend klassifizieren zu können.	-
	Interesse	Rauminformation	Als Lehrkraft muss ich den einfachen Zugriff auf Funktionen des Systems verfügbar haben um das System mit möglichst wenig Aufwand auf meiner Seite nutzen zu können.	-
	Interesse	Schnittstelle des Systems	Als Lehrkraft muss ich wissen welche Informationen ich in das System eintragen muss um das System effizient nutzen zu können	-
	Anrecht	Reservierung des Raumes	Als Lehrkraft muss ich das alleinige Recht auf eine Reservierung für einen mir vorgeschlagenen Raum verfügbar haben um meine Arbeitsaufgabe effizient zu erledigen.	-
	Anrecht	Belegung des Raumes	Als Lehrkraft muss ich das alleinige Recht auf die Buchung eines von mir Reservierten Raumes verfügbar haben um meine Arbeitsaufgabe effizient zu erledigen.	-

Tabelle 15: Stakeholderanalyse

Bezeichnung	Beziehung zum System	Objektbereich	Erfordernis, Erwartung	Prio.
Lehrkraft	Anspruch	Schnittstelle des Systems	Als Lehrkraft muss ich wissen das ich mit dem System einen Raum effizienter finden kann um mich für die Nutzung des Systems entscheiden zu können.	-
	Anspruch	Reservierung des Raumes	Als Lehrkraft muss ich die Bevorzugung meiner Buchung gegenüber Studenten verfügbar haben um den Veranstaltungsfluss ohne weitere unterbrechungen fortführen zu können.	-
Lerner	Anspruch	Rauminformation	Als Lerner muss ich die korrekten Rauminformationen verfügbar haben um einen Raum als für meine Arbeitsaufgabe passend klassifizieren zu können.	-
	Interesse	Rauminformation	Als Lerner muss ich den einfachen Zugriff auf Funktionen des Systems verfügbar haben um das System mit möglichst wenig Aufwand auf meiner Seite nutzen zu können.	-
	Interesse	Schnittstelle des Systems	Als Lerner muss ich wissen welche Informationen ich in das System eintragen muss um das System effizient nutzen zu können	-

Tabelle 16: Stakeholderanalyse

Bezeichnung	Beziehung zum System	Objektbereich	Erfordernis, Erwartung	Prio.
Lerner	Anrecht	Reservierung des Raumes	Als Lerner muss ich das alleinige Recht auf eine Reservierung für einen mir vorgeschlagenen Raum verfügbar haben um meine Arbeitsaufgabe effizient zu erledigen.	-
	Anrecht	Reservierung des Raumes	Als Lerner muss ich das alleinige Recht auf die Buchung eines von mir Reservierten Raumes verfügbar haben um meine Arbeitsaufgabe effizient zu erledigen.	-
	Anspruch	Schnittstelle des Systems	Als Lerner muss ich wissen das ich mit dem System einen Raum effizienter finden kann um mich für die Nutzung des Systems entscheiden zu können.	-
Wissenschaftliche Mitarbeiter	Anspruch	Rauminformation	Als wissenschaftlicher Mitarbeiter muss ich die korrekten Rauminformationen verfügbar haben um einen Raum als für meine Arbeitsaufgabe passend klassifizieren zu können.	-

Tabelle 17: Stakeholderanalyse

Bezeichnung	Beziehung zum System	Objektbereich	Erfordernis, Erwartung	Prio.
Wissenschaftliche Mitarbeiter	Interesse	Rauminformation	Als wissenschaftlicher Mitarbeiter muss ich den einfachen Zugriff auf Funktionen des Systems verfügbar haben um das System mit möglichst wenig Aufwand auf meiner Seite nutzen zu können.	-
	Interesse	Schnittstelle des Systems	Als wissenschaftlicher Mitarbeiter muss ich wissen welche Informationen ich in das System eintragen muss um das System effizient nutzen zu können	-
	Anrecht	Reservierung des Raumes	Als wissenschaftlicher Mitarbeiter muss ich das alleinige Recht auf eine Reservierung für einen mir vorgeschlagenen Raum verfügbar haben um meine Arbeitsaufgabe effizient zu erledigen.	-
	Anrecht	Belegung des Raumes	Als wissenschaftlicher Mitarbeiter muss ich das alleinige Recht auf die Buchung eines von mir Reservierten Raumes verfügbar haben um meine Arbeitsaufgabe effizient zu erledigen.	-

Tabelle 18: Stakeholderanalyse

Bezeichnung	Beziehung zum System	Objektbereich	Erfordernis, Erwartung	Prio.
Wissenschaftliche Mitarbeiter	Anspruch	Schnittstelle des Systems	Als wissenschaftlicher Mitarbeiter muss ich wissen das ich mit dem System einen Raum effizienter finden kann um mich für die Nutzung des Systems entscheiden zu können.	-
Institutsverwaltung	Anspruch	Rauminformation	Als Institutsverwaltung muss ich die korrekten Rauminformationen verfügbar haben um einen Raum als für meine Arbeitsaufgabe passend klassifizieren zu können.	-
	Interesse	Rauminformation	Als Institutsverwaltung muss ich den einfachen Zugriff auf Funktionen des Systems verfügbar haben um das System mit möglichst wenig Aufwand auf meiner Seite nutzen zu können.	-
	Interesse	Schnittstelle des Systems	Als Institutsverwaltung muss ich wissen welche Informationen ich in das System eintragen muss um das System effizient nutzen zu können	-

Tabelle 19: Stakeholderanalyse

Bezeichnung	Beziehung zum System	Objektbereich	Erfordernis, Erwartung	Prio.
Institutsverwaltung	Anrecht	Reservierung des Raumes	Als Institutsverwaltung muss ich das alleinige Recht auf eine Reservierung für einen mir vorgeschlagenen Raum verfügbar haben um meine Arbeitsaufgabe effizient zu erledigen.	-
	Anrecht	Belegung des Raumes	Als Institutsverwaltung muss ich das alleinige Recht auf die Buchung eines von mir reservierten Raumes verfügbar haben um meine Arbeitsaufgabe effizient zu erledigen.	-
	Anspruch	Schnittstelle des Systems	Als Institutsverwaltung muss ich wissen das ich mit dem System einen Raum effizienter finden kann um mich für die Nutzung des Systems entscheiden zu können.	-
Administrator	Anspruch	System	Als Administrator muss ich die Möglichkeit der Wartung des Systems verfügbar haben um auf Institutspezifische Probleme und Wünsche eingehen zu können.	-

Tabelle 20: Stakeholderanalyse

Bezeichnung	Beziehung zum System	Objektbereich	Erfordernis, Erwartung	Prio.
Administrator	Anspruch	Rauminformation	Als Administrator muss ich eine Möglichkeit zum Hinzufügen, Bearbeiten und Löschen von Räumen und ihren Informationen verfügbar haben um das System flexibel an das Institut anpassen zu können.	-
	Interesse	Rauminformation	Als Administrator muss ich den einfachen Zugriff auf Rauminformationen verfügbar haben um diese auf ihre Korrektheit überprüfen	-
	Anteil	System	Als Administrator muss ich wissen das alle von mir erstellten Inhalte des Systems auch mir gehören um das System ohne Bedenken warten zu können.	-
Institut	Anspruch	organisationsinternen Informationen	Als Institut muss ich wissen das Informationen über das Institut oder Teile des Instituts nicht an außenstehende weitergegeben werden um mich für die Nutzung des Systems entscheiden zu können.	-

Tabelle 21: Stakeholderanalyse

Bezeichnung	Beziehung zum System	Objektbereich	Erfordernis, Erwartung	Prio.
Institut	Anspruch	Veranstaltungen(fixe Daten)	Als Institut muss ich wissen das wöchentlich wiederkehrende Veranstaltungen in das System integriert werden können um den reibungslosen Ablauf dieser garantieren zu können.	-
	Interesse	System	Als Institut muss ich wissen das das System aktiv genutzt wird um mit dem System effektiv Veranstaltungen planen zu können.	-
	Anrecht	System	Als Institut muss ich wissen das das System fehlerfrei betrieben werden kann um mich für die Nutzung/Einbindung des Systems im Institut entscheiden zu können.	-
	Anrecht	System	Als Institut muss ich die dauerhafte Erreichbarkeit des Systems verfügbar haben um den reibungslosen Ablauf von Veranstaltungen garantieren zu können.	-