

Digitaler Campus

Audit 2 im Entwicklungsprojekt im WS2223



Inhaltsverzeichnis

Iteration Audit 1

1. Problemstellung
2. Zielsetzung
3. Leitthema
4. Stakeholderanalyse
5. Domänenmodell
6. Alleinstellungsmerkmale

Audit 2

1. Projektrisiken
2. Spezifikation der Proof-of-Concepts
3. Projektplan Audit 2
4. Projektplan Audit 3



Iteration Audit 1



Problemstellung

Abbildung des Campus

Studentische Einrichtungen wie der Campus Gummersbach verfügen oft über mehrere Gebäude und Räume, die nicht immer eindeutig ausgezeichnet sind. Oft hat man keinen Überblick darüber, welche Funktion ein Raum hat, welches Material zur Verfügung steht und ob ein Raum bereits belegt ist.

Einige Hochschulen und Universitäten bezeichnen sich selbst als unübersichtlich bzw. ist es bekannt, dass besonders Erstsemester sich nicht auf ihrem Campus zurechtfinden. Auch im HochschulPlanungsSystem (HoPS) der TH Köln sind diese Informationen nur oberflächlich zu finden. Der 27-seitigen Erstsemesterbroschüre ist nur eine Seite dem Campus gewidmet, wo dieser nur grob kategorisiert wird.

Was wurde iteriert

- Problemstellung spezifiziert
- Fokus auf Raumverfügbarkeit und Raummaterial



Zielsetzung

Vision

Das Ziel ist es, eine Abbildung des Campus in digitaler Form darzustellen. Hierbei soll es Benutzern möglich sein, nach Räumen und Equipment zu filtern. Zum Equipment gehören z. B. die benötigte Software, Hardware oder die verfügbaren Sitzplätze. Die Raumverfügbarkeit soll auf Basis aktueller Stundenpläne, Sensoren und/oder Benutzereingaben ermittelt werden.

Zusätzlich zur Filterfunktion können sich Nutzer auch direkt Informationen und Metadaten zu individuellen Räumen anzeigen lassen.

Was wurde iteriert

- Technisch unabhängig spezifiziert
- Fokus auf Raumverfügbarkeit und Raummaterial



Leitthema

Synchronisation

Zur Einbindung einer Multiscreen Experience wird die Synchronisation von Raumbelagungen in den Fokus gestellt. Hierfür können Nutzer beim Betreten von Räumen in der Anwendung angeben (z. B. manuelle Eingabe oder Scannen von Codes), dass sie im Raum sind und beim Verlassen, dass sie den Raum verlassen haben. Die gemessenen Daten sollen direkt in der Anwendung zur Verfügung stehen und somit die Benutzer und Informationssysteme vor Räumen informieren.

Device Shifting

Es werden begrenzt Rauminformationen auf einem Display angezeigt. Mittels NFC/Code kann man sich weitere Informationen auf dem Smartphone anzeigen lassen.

Was wurde iteriert

- Synchronisation auf Zielsetzung angepasst
- Device Shifting hinzugefügt



Stakeholderanalyse

Bezeichnung		Bezug zum System	Objektbereich	Erfordernis / Erwartung
Benutzer	Studierende Wissenschaftliche Mitarbeiter Studentische Hilfskraft Pförtner	Anteil	Nutzer des Systems	Als Benutzer muss ich wissen das dieses System existiert, um es nutzen zu können.
		Interesse	Nutzer des Systems	Als Benutzer muss man einen Überblick über die Räume der Hochschule haben, um effizient verfügbare Räume finden zu können.
		Interesse	Nutzer des Systems	Als Benutzer muss man einen Überblick über die Räume der Hochschule haben, um effizient ein benötigtes Material finden zu können.
		Anspruch	Nutzer des Systems	Als Benutzer müssen die Metadaten zuverlässig sein.
		Anspruch	Nutzer des Systems	Als Benutzer müssen die Daten bezüglich der Raumverfügbarkeit zuverlässig sein.
		Anspruch	Nutzer des Systems	Als Benutzer müssen die Daten bezüglich der Raumverfügbarkeit stets aktuell (synchronisiert) sein.
	Dozent	Anspruch	Nutzer des Systems	Als Benutzer muss die Interaktion zur Bestimmung der Raumverfügbarkeit angenehm, schnell und minimal sein, um die Anwendung ungestört nutzen zu können.
Hochschule		Interesse	Nutzer des Systems	Als Dozent muss man einen Überblick über die Räume der Hochschule haben, um bei kurzfristigen Raumänderungen einen verfügbaren Raum finden zu können.
		Anspruch	Bereitstellung von Informationen	Als Hochschule muss man Informationen zu den Räumen haben, um diese bereitstellen zu können
Besucher		Anspruch	Förderer	Als Hochschule muss das System genutzt werden, um den Weiterbetrieb des Systems zu begründen
		Interesse	Besucher des Campus	Als Besucher muss man einen Überblick über die Räume der Hochschule haben, um effizient einen gesuchten Raum finden zu können.
primärer Stakeholder				
sekundärer Stakeholder				
tertiärer Stakeholder				

Für Audit 2 wurde eine Stakeholderanalyse durchgeführt.

Folgende Stakeholder sind dabei als primäre Stakeholder festzuhalten:

Benutzer, die das System primär nutzen und dabei folgende als wichtig einzustufende Erfordernisse/Erwartungen haben:

Als Benutzer muss man einen Überblick über die Räume der Hochschule haben, um effizient verfügbare Räume finden zu können.

Als Benutzer muss man einen Überblick über die Räume der Hochschule haben, um effizient ein benötigtes Material finden zu können.

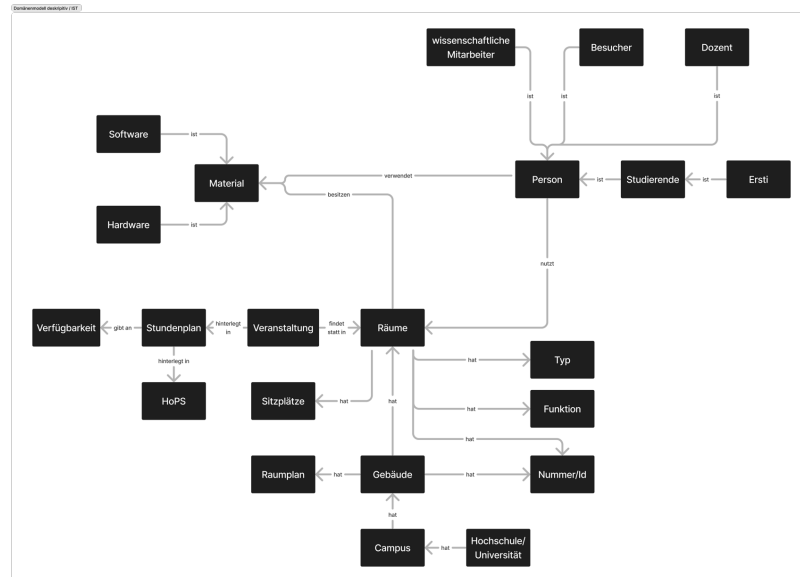
Als Benutzer muss die Interaktion zur Bestimmung der Raumverfügbarkeit angenehm, schnell und minimal sein, um die Anwendung ungestört nutzen zu können.

Die Bezeichnung des Benutzers stellt dabei eine Generalisierung einzelner Benutzerbezeichnungen dar:

- Studierende
- Wissenschaftliche Mitarbeiter
- Studentische Hilfskräfte
- Pförtner

Die Hochschule wird als sekundärer Stakeholder festgehalten.

Domänenmodell Deskriptiv



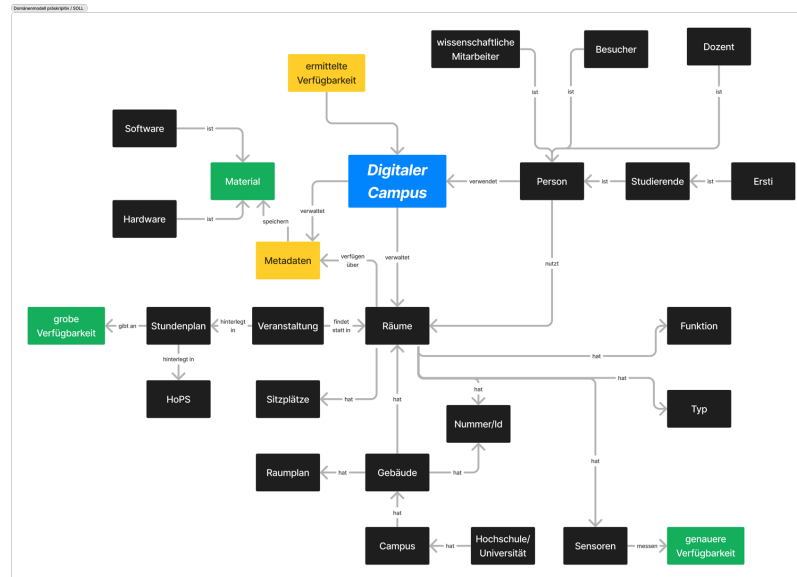
Im Domänenmodell werden die Stakeholder des Systems gemäß ihrer Beziehungen zueinander dargestellt.

Hierbei ist in der deskriptiven Modellierung zu erkennen, dass Personen Räume und die in den Räumen zur Verfügung stehenden Materialien nutzen. Einer Person ist dabei nicht klar, welche Materialien in welchen Räumen bereitstehen und welche Räume verfügbar sind.

Iteration:

- Wissenschaftliche Mitarbeiter wurden hinzugefügt
- Dozent wurde hinzugefügt
- Räumen wurden Typen und Nummern/Ids
- Gebäuden wurden Nummern/IDs

Domänenmodell Präskriptiv



Im präskriptiven Domänenmodell werden die Beziehungen der Stakeholder im Zusammenhang zum erstellenden System dargestellt.

Zu erkennen ist, dass die Anwendung, der „Digitale Campus“, die zentrale Schnittstelle zwischen den Personen, den benötigten Materialien und den jeweiligen Räumen darstellt.

Nutzer sollen die zur Verfügung stehenden Metadaten der Räume direkt über das System einsehen können.

Iteration:

- Wissenschaftliche Mitarbeiter wurden hinzugefügt
- Dozent wurde hinzugefügt
- Räumen wurden Typen und Nummern/Ids
- Gebäuden wurden Nummern/Ids
- Sensoren zur Raumverfügbarkeit und Materialien für Rauminformationen hinzugefügt (grün)
- Metadaten und die erweiterte Verfügbarkeit, welche Nutzern angezeigt werden (gelb)
- Das System (Digitaler Campus) im Zentrum (blau)

Audit 2



Alleinstellungsmerkmale

Physischer Raumplan

- Größere Datentiefe (mehr Informationen können zu einem Raum dargestellt werden)
- Interaktivität nicht gegeben
- keine Informationen zur Raumverfügbarkeit vorhanden

HoPS Raumplan

- Interaktivität nicht gegeben
- Übersichtlichkeit nicht gegeben
- keine aktuellen Informationen zur Raumverfügbarkeit vorhanden

HoPS Ausstattung

- Interaktivität nicht gegeben
- Übersichtlichkeit nicht gegeben
- Ausstattung in HoPS nicht detailliert genug
- Keine intuitive Suche/Filterung nach Ausstattung in HoPS

Vorteil gegenüber mazemap(<https://www.mazemap.com/>):

- Spezialisierung auf Campus-Umfeld
- Integration Raumverfügbarkeit
- Integration Rauminformationen (Equipment)

Gebäudepläne(<https://davinci.stueber.de/floorplan.php>):

- Ähnliches System muss für den Campus erstellt werden.



Projektrisiken Part 1

Projektumfang

- Projektumfang ungenau definiert
- Schleichende Erweiterungen die den Projektumfang sprengen

Änderungsmanagement

- Unnötige Refaktorisierung von Code
- Änderungen werden falsch priorisiert

Kommunikation

- Terminabstimmung nicht immer möglich
- Zu viel Kommunikation
- Zu wenig Kommunikation

Ressourcen und Projektteam / Beschaffung

- Erforderliche Daten sind nicht zugänglich
- Lernkurven zu hoch bzw. Erfahrung zu niedrig; neue Technologien erfordern zu viel Zeit zur Einarbeitung

System-Architektur

- gewünschte Architektur des Systems kann nicht wie geplant umgesetzt werden
- Architektur wird zu unflexibel entworfen



Projektrisiken Part 2

Produkt-Design

- Das Design ist nicht intuitiv genug und wird nicht benutzt
- Die Darstellung ist nicht eindeutig. Anwender finden sich nicht zurecht
- Zu viele UI Komponenten
- Die Darstellung des Campus mit WebGL kann die Performance beeinträchtigen
- Anwendung ist entspricht nicht den W3C Accessibility Guidelines (WCAG)

Technisches

- Messung der Raumverfügbarkeit ist zu ungenau (Untersuchung der jeweiligen Möglichkeiten zur Bestimmung der Verfügbarkeit ist erforderlich)
- Eingesetzte Technologien sind nicht für den Anwendungsfall geeignet
- Technologische Funktionen sind nicht skalierbar oder performant genug
- Informationssicherheit wird nicht genug beachtet
- Technologische Funktionen des Systems werden nicht dokumentiert
- Technologische Funktionen des Systems sind nicht oder nur schwer erweiterbar
- Zu viele Ressourcen/Skripte müssen geladen werden

Integration

- Vorhandene Systeme (z.B. Transpondersystem) lassen sich nicht verwenden

Anforderungen

- Die Anforderungen wurden falsch/ungenau spezifiziert

Externe Risiken

- Modulziele werden verfehlt
- Produkte zum Testen der PoCs sind nicht verfügbar
- Die Hochschule verbietet das Projekt



Proof of Concept Inhalte

Inhaltsverzeichnis PoCs:

- Architektur - Push-Benachrichtigungen
- Architektur - Kamera
- Architektur – Standortfreigabe

- Verknüpfung von Daten und Digitaler Abbildung
- Digitale Darstellung Campus Gummersbach

- Raumverfügbarkeit - NFC
- Raumverfügbarkeit - Button / Touch-Display
- Raumverfügbarkeit - HoPS Scraping
- Raumverfügbarkeit - Laserschranke
- Raumverfügbarkeit - Nutzungsbereitschaft verschiedener Methoden zur Messung
- Raumverfügbarkeit - Stundenplan
- Raumverfügbarkeit - GPS
- Raumverfügbarkeit - InkPaper Display
- Raumverfügbarkeit - Bewegungsmelder
- Raumverfügbarkeit - Nutzung Transponder Infrastruktur



Proof-of-Concept: Architektur - Push-Benachrichtigungen

Beschreibung

Es muss überprüft werden, ob per standardmäßigen Webtechnologien Push-Benachrichtigungen an einen Nutzer gesendet werden können.

Exit-Kriterien

- Die Website kann NutzerInnen durch Push-Benachrichtigungen zur Interaktion mit dem System auffordern.

Fail-Kriterien

- Die Website kann NutzerInnen NICHT durch Benachrichtigungen zur Interaktion mit dem System auffordern.

Fallbacks

- Benachrichtigungen auf das Gerät der NutzerInnen könnten durch SMS/E-Mails ersetzt werden.



Proof-of-Concept: Architektur - Kamera

Beschreibung

Es muss überprüft werden, ob per standardmäßigen Webtechnologien über die Kamera eines Nutzers ein QR-Code oder ähnliches eingelesen werden kann.

Exit-Kriterien

- NutzerInnen können direkt über die Website über eine integrierte Kamera etwas Scannen.

Fail-Kriterien

- NutzerInnen können NICHT direkt über die Website über eine integrierte Kamera etwas Scannen.

Fallbacks

- Wenn der Zugriff auf die Kamera der NutzerInnen nicht erlaubt wird, könnte in diesem Fall entweder kein Scannen erforderlich sein; z.B. könnte man an den Displays vor einem Raum einen (täglich?) generierten Code anzeigen, welcher in der Anwendung eingegeben werden muss. Ansonsten wäre ein Dateiupload denkbar.



Proof-of-Concept: Architektur - Standortfreigabe

Beschreibung

Es muss überprüft werden, ob per standardmäßigen Webtechnologien der Standort eines Nutzers freigegeben werden kann.

Exit-Kriterien

- NutzerInnen können direkt über die Website ihren Standort freigeben.

Fail-Kriterien

- NutzerInnen können NICHT über die Website ihren Standort freigeben.

Fallbacks

- Standortfreigaben könnten durch NutzerInnen direkt in der Anwendung erfolgen und es müssten keine APIs (z.B. GPS) verwendet werden (z. B. durch Auswahl auf einer Karte)



Proof-of-Concept: Verknüpfung von Daten und Digitaler Abbildung

Beschreibung

Wie können Metadaten eines Raumes (Sitzplätze, vorhandene Hardware) am effektivsten mit einem Digitalen Raum verknüpft werden.

Exit-Kriterien

2D

- Daten lassen sich unbeschränkt in der Vektor Datei einbinden.

3D

- Bei der Erstellung der glTF Datei ist es möglich Räumen und Gebäuden mit IDs zu verknüpfen.

2.5D

- Das Schema der JSON Datei entspricht den Anforderungen an das System.

Fail-Kriterien

2D

- Daten lassen sich nur beschränkt in der SVG Datei einbinden.

3D

- Es ist nicht Möglich Räume und Gebäude in eigene Komponenten zu gliedern und zu benennen.

2.5D

- Es lässt sich kein einheitliches Schema für die Darstellung der Daten finden.

Fallbacks

Durch die Planung von verschiedenen Darstellungsmöglichkeiten sind Fallbacks automatisch gegeben. Eine 2.5D Darstellung eignet sich im Vorfeld am Besten zur Verknüpfung von Daten und Komponenten, da alles in einer JSON Datei gespeichert ist.



Proof-of-Concept: Digitale Darstellung Campus Gummersbach

Beschreibung

Es soll geprüft werden welche Darstellung sich am Besten für den Campus Gummersbach eignet. Zur Implementierung soll ThreeJS verwendet werden.

- 2D: Die Abbildung wird ausschließlich in 2D dargestellt.
- 3D: Für die Abbildung wird ein vollwertiges 3D-Modell (.glTF) des Campus Gummersbach verwendet.
- 2.5D: Als Vorlage dient eine Vektor-Grafik oder Json Datei, die den Campus Gummersbach darstellen soll. Durch Extrudieren und anderen Methoden wird daraus eine drei dimensional Lösung erstellt.

Exit-Kriterien

- 2D
- Die Vektor Grafik lässt sich problemlos mit ThreeJS einbinden.
 - Möglichkeiten zur Interaktion sind gegeben.
- 3D
- Durch die Zusammenarbeit mit einem anderen Team kann das 3D Modell zur Verfügung gestellt werden.
 - Das gestellte 3D Modell entspricht den selben Anforderungen.
- 2.5D
- Das Aufbereiten der Vektor Grafik, damit ein 3D dimensionales Ergebnis erzielt wird, funktioniert zuverlässig.
 - Die generierte Lösung lässt sich interaktiv verwenden.

Fail-Kriterien

- 2D
- Die Grafik lässt sich nicht einbinden.
 - Die Grafik lässt sich einbinden, bietet allerdings nicht ausreichend Möglichkeiten zur Interaktion.
- 3D
- Das erhaltene Modell entspricht nicht unseren Anforderungen.
 - Die eigene Erstellung des Modells ist zu Aufwendig.
 - Die Erwartung an das Modell entspricht nicht den Fähigkeiten des Erstellers.
- 2.5D
- Eine generische Lösung ist nur begrenzt möglich.

Fallbacks

- Es findet sich eine bessere Lösung als ThreeJS zum rendern der Abbildung. (2D)
- Eine Darstellung (2D, 3D, 2.5D) erweist sich als wesentlich intuitiver und simpler zu implementieren.



Proof-of-Concept: Raumverfügbarkeit - NFC

Beschreibung

Um den Aufwand für NutzerInnen gering zu halten, soll die Möglichkeit zum Scannen von NFC Daten vor einem Raum untersucht werden. NutzerInnen sollen durch das Scannen direkt auf die Informationsseite zu einem Raum, bzw. zur Eingabe der Raumverfügbarkeit geleitet werden.

Exit-Kriterien

- NutzerInnen können über ein mobiles Gerät per NFC vor einem Raum Daten einscannen und gelangen somit direkt auf die Informationsseite eines Raumes
- Wenn erwünscht, kann direkt angegeben werden, dass man den Raum betritt

Fall-Kriterien

- Die NFC Daten können nicht von NutzerInnen eingescannt werden.
- Die eingelesenen Daten sind falsch, veraltet oder ungültig/inkonsistent.

Fallbacks

- Manuelle Eingabe der Raumnummer
- (Scannen eines QR-Codes durch NutzerInnen)
- (Eingabe eines Codes vom Display)



Proof-of-Concept: Raumverfügbarkeit - Button / Touch-Display

Beschreibung

NutzerInnen können über einen Button vor einem Raum angeben, dass sie den Raum betreten. Durch drücken dieses Buttons wird ein Signal / eine Nachricht an das Backend geschickt.

Exit-Kriterien

- Der Button kann erfolgreich gedrückt werden, das Backend erhält diese Information und die Nutzeranzahl wird erhöht.

Fail-Kriterien

- Der Button funktioniert nicht und die Nutzeranzahl kann nicht erfolgreich erhöht werden.
- Das Signal kommt nicht beim Backend an
- Der Button steckt fest (vielleicht klebt er?)

Fallbacks

- Nutzer kann über die Anwendung angeben, wenn ein Raum betreten wird



Proof-of-Concept: Raumverfügbarkeit - HoPS Scraping

Beschreibung

Das HoPS soll gescraped werden um die Stundenpläne pro Raum in eine für uns nutzbare Form von Daten umzuwandeln.

Exit-Kriterien

- Wir haben die Daten erfolgreich gescraped und abgespeichert.

Fail-Kriterien

- Das scrapen klappt aus jeglichen Gründen nicht.

Fallbacks

- (1) Die Betreiber des HoPS nach den Daten fragen und diese dann transformieren.



Proof-of-Concept: Raumverfügbarkeit - Laserschranke

Beschreibung

Zur Messung der Raumverfügbarkeit wird der Einsatz einer Laserschranke als Sensor in Betracht gezogen.
Die Genauigkeit der Messung soll hierbei untersucht werden.

Exit-Kriterien

- Die Schranke kann zuverlässig messen, wie viele Personen einen Raum betreten oder verlassen.
- Die Messung ist ungenau, aber noch in einem Rahmen der in Ordnung wäre. (1-2 Personen ?)

Fail-Kriterien

- Die Messung des Sensors ist zu ungenau.

Fallbacks

- Eine Mischung aus mehreren Sensoren kann zur Messung der Raumverfügbarkeit verwendet werden. (z. B. Bewegungsmelder)



Proof-of-Concept: Raumverfügbarkeit - Nutzungsbereitschaft verschiedener Methoden zur Messung

Beschreibung

Die Raumverfügbarkeit bzw. die Anzahl an Personen in einem Raum soll anhand einer von drei Methoden, welche unterschiedlich viel Benutzer-Interaktion benötigen, festgestellt werden:

- GPS (benötigt Erlaubnis der Nutzer)
- Nutzereingabe innerhalb des Systems bei betreten / verlassen eines Raums.
- Scannen eines QR-Codes am Eingang eines Raums.

Exit-Kriterien

Methode 1:

- Der Nutzer erlaubt das Tracking mittels GPS auf dem Hochschulgelände.

Methode 2:

- Der Nutzer teilt uns mittels betätigen eines Buttons mit, das dieser den Raum betreten hat.

Methode 3:

- Der Nutzer scannt den QR code vor betreten des Raums.

Fail-Kriterien

- Der Nutzer nutzt keine dieser Möglichkeiten.

Fallbacks

- (1) Wir nutzen Bewegungsmelder um zu messen wann eine Person einen Raum betritt / verlässt.



Proof-of-Concept: Raumverfügbarkeit - Stundenplan

Beschreibung

Unser System soll anhand der Daten des Stundenplans die Raumverfügbarkeit bestimmen.

Exit-Kriterien

- Das System erhält korrekte Daten welche der Realität entsprechen.
- Wir haben Zugriff auf die Daten.

Fall-Kriterien

- Ein Dozent ändert kurzfristig den Raum.
- Der Stundenplan ändert sich und wird nicht im PSSO aktualisiert.
- Wir haben keinen Zugriff auf die Daten.

Fallbacks

- (1) Falls ein Dozent den Raum ändert, wird dies wahrscheinlich im internen Raumbuchungstool hinterlegt sein, wir sollten auf Veränderungen reagieren und unsere Daten anpassen.



Proof-of-Concept: Raumverfügbarkeit - GPS

Beschreibung

GPS soll genutzt werden um dem System mitzuteilen in welchem Raum sich ein Benutzer befindet.
- Nutzungsbereitschaft GPS zu benutzen wird vorausgesetzt.

Exit-Kriterien

- Der Standort welches das GPS uns mitteilt entspricht dem reellen Standort der Person.

Fail-Kriterien

- Die Person befindet sich in einem anderen Raum als uns angezeigt wird.

Fallbacks

- (1) Wird die Person in einem Spektrum geortet wo nur ein Raum existiert für den wir die Raumverfügbarkeit tracken, wird diese Person diesem Raum automatisch zugeordnet (Worst Case Assumption Method)



Proof-of-Concept: Raumverfügbarkeit - InkPaper Display

Beschreibung

Ein Inkpaper - Display soll vor einem Raum platziert werden, Daten aus dem Backend erhalten und diese anzeigen.

Exit-Kriterien

- Die Daten werden erfolgreich empfangen.
- Die Daten werden angezeigt.

Fail-Kriterien

- Das Display wird verdeckt.
- Die Daten auf dem Display sind veraltet.
- Display geht aus.

Fallbacks

(2/3) Alle Displays, bzw. die Microcontroller an denen diese angeschlossen sind werden regelmäßig angepingt um sicherzustellen, dass eine konstante Verbindung vorhanden ist.



Proof-of-Concept: Raumverfügbarkeit - Bewegungsmelder

Beschreibung

Ein Bewegungsmelder soll unmittelbar neben den Eingang eines Raumes platziert werden, um zu erkennen wenn eine Person den Raum betritt/verlässt.

Exit-Kriterien

- Der Bewegungsmelder erkennt dass eine Person den Raum betreten hat.

Fail-Kriterien

1. Der Bewegungsmelder erkennt nicht, dass eine Person den Raum betreten/verlassen hat.
2. Der Bewegungsmelder meldet nur eine Person obwohl eine Gruppe den Raum gleichzeitig betreten hat.
3. Der Bewegungsmelder erkennt, dass eine Person den Raum betritt, obwohl diese den Raum gerade verlässt.
4. Der Bewegungsmelder wird blockiert.
5. Der Bewegungsmelder wird aus bössartiger Absicht mehrmals (z.B. mit der Hand) ausgelöst.

Fallbacks

- (1) Nach einem bestimmten Zeitabschnitt wird die Anzahl der Personen im Raum dekrementiert, auch wenn der Sensor kein Verlassen gemeldet hat
- (2) Einen Algorithmus implementieren, welcher je nach Länge des Auslösens einen Wert errechnet mit der wahrscheinlichen Anzahl an Personen
- (2/3/5) Einen Zweiten Sensor platzieren (Double Beam Approach), das Exit-Kriterium tritt nur ein wenn beide Sensoren ausgelöst wurden
- (5) Nach dem auslösen den Bewegungsmelder eine gewisse Auszeit bis zur nächsten Messung geben. (allerdings erhöht dies die Wahrscheinlichkeit für Fail-Kriterium 2)



Proof-of-Concept: Raumverfügbarkeit - Nutzung Transponder Infrastruktur

Beschreibung

Es soll geprüft werden, ob eine Schnittstelle mit dem vorhandenen Transponder System für die automatische Schließung und Öffnung der Türen am Campus Gummersbach vorhanden oder möglich wäre.

Exit-Kriterien

1. Schnittstelle ist vorhanden.
2. Es ist möglich auf die Schnittstelle zuzugreifen.
3. Schnittstelle bietet Auskunft über den State der Türen. Ist ein Raum geöffnet oder geschlossen.

Fail-Kriterien

- Es gibt keine Schnittstelle.
- Es gibt eine Schnittstelle, liefert aber nicht die gewünschten Daten.
- Es gibt eine Schnittstelle, die gelieferten Daten sind allerdings unvollständig.
- Es gibt eine Schnittstelle, der Zugang ist nicht für das Projekt zugänglich.

Fallbacks

- Es muss nicht auf dem vorhandenen Transponder System zugegriffen werden. Es reicht aus einen Prototyp und Mock-Daten zu verwenden.
- Aktuelles Transponder-System bietet keine Schnittstelle. Gibt es vergleichbare Systeme die eine Schnittstelle mit den erwünschten Anforderungen bereitstellen?



Projektplan Audit 2

Projektplan Audit 3

Audit 3 8 Dec 12 - Jan 12, 2023									
35	<input type="radio"/> Erster vertikaler Rapid Prototype #56	-	Audit 3	-			<input type="button" value="New"/>	-	Audit 3
36	<input checked="" type="radio"/> Durchfuehrung der PoCs #57		Audit 3				<input type="button" value="New"/>		Audit 3
37	<input type="radio"/> Modellierungen iterieren #58	-	Audit 3	-			<input type="button" value="New"/>	-	Audit 3
38	<input type="radio"/> Modellierung der Anwendungstgik #59	-	Audit 3	-			<input type="button" value="New"/>	-	Audit 3
39	<input type="radio"/> Deliverables für den 4. Audit (Projektplan) #60	-	Audit 3	-			<input type="button" value="New"/>	-	Audit 3
40	<input type="radio"/> Fachperspektiven-Spezifische Leitfragen #43	-	Audit 3	-			<input type="button" value="New"/>	-	Audit 3
41	<input type="radio"/> Erfordernisse #64	-	Audit 3	-			<input type="button" value="New"/>	-	Audit 3
42	<input type="radio"/> Anforderungen #65	-	Audit 3	-			<input type="button" value="New"/>	-	Audit 3

