

Gesamtspezifikation

rlm1-04: Interactive Room Map Display



Studiengang: Informatik

Autor: Marc Rudolf Fuhrer, Nicolas Anthony Waser, Nicola Sacha Eigel

Betreuer: Prof. Dr. Michael Röthlin

Datum: 20.01.2023

Inhaltsverzeichnis

1	Executive Summary	4
2	Versionshistorie	5
3	Stakeholder und Projektteam	6
	3.1 Primäre Stakeholder	6
	3.2 Sekundäre Stakeholder	7
4	Systemabgrenzung	8
	4.1 Interactive Room Map Display	8
	4.2 Database	8
	4.3 REST-API	8
	4.4 Publisher	8
	4.5 Subscriber	8
	4.6 Webserver	8
	4.7 Grafik	9
5	Anforderungen	10
	5.1 Quellen und Vorgehen	10
	5.1.1 Quellen	10
	5.1.2 Vorgehen	10
	5.2 Funktionale Anforderungen	11
	5.2.1 Low-Level Use Case Diagramm	11
	5.2.2 High-Level Use Case-Diagramm	12
	5.2.3 User Stories (Mindset High Level)	13
	5.2.4 User Stories (Mindset Low Level)	20
	5.3 Nicht-funktionale Anforderungen	25
	5.3.1 Reliability	25
	5.3.2 Ease of use	25
	5.3.3 Speed	25
	5.3.4 Robustness	25
	5.3.5 Portability	25
	5.3.6 Size	25
	5.3.7 Integrity	25
	5.4 Domänenanforderungen	26
6	Domänenmodell	27
7	Entity Relationship Diagram	28
8	DB Sicherheit	29
	8.1 Zugriffskontrolle	29
	8.2 Authentifizierung	29
	8.3 Verschlüsselung	29
	8.4 Konsistenz	29
	8.5 Datensicherung	29
9	ACL Konzept	30
10	Sicherheitskonzept REST-API	31
	10.1 Generell	31
	10.2 Authentifizierungsmethode	31
	10.3 Verschlüsselung	31
	10.4 Quotas and Throttling	31
	10.5 Contenttype Validation	31
	10.6 Input Validation	31
	10.7 Error Handling	32
	10.8 Response Headers	32
11	Ablauf Publishing-Prozess	33

12 Literaturverzeichnis	34
12.1 Quellen betreffend (Nicht-) Funktionalen Anforderungen	34
12.1.1 Literatur FA/NFA	34
12.2 Quellen zu Sicherheitskonzept REST-API	34
13 Anhang	35
13.1 Anhänge bzgl. Anforderungen	35
13.1.1 Project Proposal rlm1-04	35
13.1.2 Project Proposal rlm1-01	36
13.1.3 Project Proposal rlm1-02	37
13.1.4 Project Proposal rlm1-05	38
13.1.5 2022-09-27 Meeting mit Herr Röthlin	39
13.1.6 2022-10-04 Meeting mit Herr Röthlin	40
13.1.7 2022-10-18 Besprechungsnotizen (Abstimmungsmeeting projektabhängiger	
Stakeholder)	41
13.1.1 2022-10-25 Meeting mit Herr Röthlin	45
13.1.1 2022-11-22 Besprechungsnotizen (Abstimmungsmeeting projektabhängiger	
Stakeholder)	48
13.2 Personas und Storyboards	50
13.2.1 Elma, BFH-Gebäude-Nutzer/BFH-Gebäude-Besucher (Low Level)	50
13.2.2 Guido, BFH-Gebäude-Besucher (High Level)	51
13.2.3 Juanito, Zukünftiges Entwicklerteam (HighLevel)	52
13.3 REST-API	53
13.3.1 Version 0.0.1	53
13.3.2 Version 1.0	56
13.3.3 Version 1.1	60
13.3.4 Version 1.2	64

1 Executive Summary

Dieses Dokument umfasst die bisherigen Resultate, welche im Rahmen des Moduls «Project 1 (BTI3031)» erarbeitet wurden. Beleuchtet wurden die Rahmenbedingungen eines realen Projekts, dessen Planung und sämtliche damit verbundenen Anforderungen.

Ziel des Dokuments ist es, einen Überblick über jene momentan vereinbarten und erarbeiteten Anforderungen des agilen Prozessverlaufs festzuhalten, die dem jeweils zuletzt bekanntem Stand entsprechen. Angefangen mit der Projektbeschreibung wurden, jeweils nach Vorbild der im Modul «Requirements Engineering (BTI11111q)» erlernten Vorgehensweise, sämtliche der Übersicht dienlichen Spezifikationen in dieses Dokument aktualisiert. Dieses Prinzip wurde so weitergeführt bis das Endresultat, also der Teil des gesamten Requirements Engineering Prozesses abgearbeitet wurde, welcher der Verschaffung einer Übersicht dienlich ist. Der Projekt-Scope wurde zu Beginn, angelehnt an die initiale Projektbeschreibung, lediglich auf das Entwickeln einer Software gelegt. Somit wurden Hardware bezogene Anforderungen aussenvorgelassen. Weiter sei zu bemerken, dass der Schwerpunkt dieses Projekts auf der agilen Prozessmanagement-Methodik lag und somit zu keinem Zeitpunkt Anspruch auf lückenlose Anforderungserhebung bestand.

Auf Grund des agilen Prozesses wurde dieses Dokument erst im 2. Sprint (20.10.2022-01.11.2022) initial erstellt, da das Projekt-Team entschied die verfügbaren Entwicklerstunden für einen bindenden 1. Entwurf der API aufzuwenden. Vorteil dieses Vorgehens war, dass sämtliche abhängige Projekt-Teams möglichst wenig Verzögerungen erleiden mussten. Mängel, welche wir im Verlauf des Projektes aufgedeckt haben, wurden rückwirkend angepasst, Modelle wurden entsprechend ergänzt resp. abgeändert.

2 Versionshistorie

Version	Datum	Autor	Änderung
1.0	31.10.2022	Alle	1. Fassung «Gesamtspezifikation - rlm1-04 -
			InteractiveRoomMapDisplay.docx»
2.0	06.12.2022	Alle	Generelle Überarbeitung nach Anforderungen «Low-Level»
3.0	20.01.2023	Alle	Aktualisierung

3 Stakeholder und Projektteam

Stakeholders wurden in zwei Gruppen unterteilt, die primären und der sekundären Stakeholder. Bei den primären Stakeholdern handelt es sich um Personen welche direkt mit dem Projekt in Kontakt kommen, währenddem es sich bei sekundären Stakeholdern um Personen handelt, die nur indirekte Berührungspunkte mit dem Projekt aufweisen.

3.1 Primäre Stakeholder

PO (Product Owner)

Der PO, der sowohl als Repräsentant der projekt-bewertenden Instanz, als auch Vertreter der Interessen des effektiven Endbenutzers und Unterhalters des Endprodukts fungiert, ist daran interessiert eine ausbaufähige Lösung zur Live-Erfassung und Darstellung der Kombination von Gebäudeplänen und verschieden gearteter Sensor-Daten als Produkt dieser Arbeit zu sehen. Er wünscht einen Service, der auf bewährten PHP-Komponenten basiert und einfach zu installieren, konfigurieren, bedienen und warten ist. Um eine gewisse User Experience zu gewährleisten, soll sowohl eine vernünftige User- und Rechteverwaltung vorhanden sein, als auch eine vereinfachte Fehlerbehandlung durch freizügiges Logging umgesetzt sein.

rlm1-04-Entwicklerteam

Das Entwicklerteammöchte eine Software-Umsetzung, die möglichst sämtliche an Sie herangetragenen Anforderungen erfüllt, oder Sie bei allfälliger Weiterführung des Projekts über den Umfang des Moduls hinaus, durch Weiterverfolgung des gewählten Pfades, zu erfüllen vermag.

Service Administrator

Der Service Administrator wünscht sowohl umfassende User- und Publisher-Konfigurationsverwaltung, als auch die Verwaltung von Beziehungen dieser beiden Teilnehmer per Konsole. Weiter soll die Subscriber-Verwaltung restriktiv über IP-Adressbereiche (inkl. Wildcard) und Authentisierung umgesetzt sein. Auch alternative Zurverfügungstellung von Inhalten für unangemeldete User soll umgesetzt sein. Des Weiteren sind Nutzungsdaten von Publishern und Subscribern zur Optimierung gefordert.

Publisher

Die Publisher sind darauf angewiesen, dass eine praktikable Schnittstelle zum Service vorzufinden ist, bei der Sie ausreichend Freiraum bzgl. vollständiger Übermittlung Ihrer Daten und deren Sicherheit haben. Realisiert wird dies über PUT-Requests via REST API per vom Service Administrator erstellte User(-Gruppen). Ordentliche http-Statusmeldungen sind unabdingbar. Sorgfältige Prüfung von Berechtigungen bzgl. Überschreiben von Daten anderer Publisher ist ebenfalls Voraussetzung. Ein Publisher möchte auch wissen, wie oft seine Daten abgerufen wurden, um seine Nutzung des Services sukzessive zu verbessern. Vorerst begnügt sich ein Publisher mit einem Low-Level-Interface, welches ihm erlaubt eine View in Form einer HTML-Seite zur Verfügung zu stellen. In einem weiteren Schritt ist ein High-Level-Interface angedacht, wie es zu Beginn des Moduls angedacht war.

Subscriber

Die Subscribers nutzen den Service, um sich die von den Publishern veröffentlichten Daten und die dazugehörigen Meta-Daten anzuzeigen. Subscribers sollen authentifiziert werden und möchten selbst bestimmen welche Informationen sie abonnieren möchten. Diese werden auf drei Arten abgerufen:

- 1. HTML, um Sie in einem Webbrowser mit selbstgewählter Freguenz als Slideshow anzuzeigen
- 2. JSON, um Sie in anderen Applikationen weiterzuverwerten
- 3. PDF als Download

3.2 Sekundäre Stakeholder

Als Sekundäre Stakeholder werden die Personen definiert welche indirekt mit dem Produkt in Kontakt kommen.

Politische Vertretung der regionalen Hochschullandschaft

Im Interesse der regionalen Bildungseinrichtungen ist eine effiziente Nutzung von Ressourcen erstrebenswert und stets als den gemeinsam definierten Zielen zuträglich einzustufen. Sowohl die Selbstwirksamkeit der Studierenden, als auch die Attraktivität des persönlichen Hinbegebens zu Begegnungsorten wird durch dieses Projekt gefördert.

BFH-Unterhalter IT-Infrastruktur

Die reibungslose Integration, sowie Wartung und Sicherstellung der Sicherheit des BFH-Netzwerks, sowie die finanzielle Tragbarkeit des Projekts sollen im dafür üblichen Rahmen gehalten werden.

4 Systemabgrenzung

In unserem Projekt liegt der Fokus auf der Validierung und Verwaltung der uns über die REST-API zur Verfügung gestellten Daten. Somit grenzen wir das Erheben und Formatieren dieser Daten ab.

4.1 Interactive Room Map Display

Die erhaltenen Daten werden auf dem Web-Service genannt «Interactive Room Map Display» veröffentlicht.

4.2 Database

Die Datenbank wird verwendet, um die für die Darstellung erforderlichen Daten abzulegen. Weiter werden darin interne Daten der Applikation, wie z. B. User Daten, gespeichert. Die Datenbank stellt das Bindeglied zwischen REST-API und dem Service dar.

4.3 REST-API

Die Schnittstelle, welche für die Interaktion mit den Publishern zuständig ist, wird in Form einer REST-API realisiert.

4.4 Publisher

Die Publisher bzw. deren Sensoren, welche uns die Informationen aus den jeweiligen Räumlichkeiten zur Verfügung stellen, werden in einem externen System eingerichtet und betrieben. Der einzige Berührungspunkt stellt hier die REST-API dar, welche ein klares Schema zum Datenaustausch vorgibt.

4.5 Subscriber

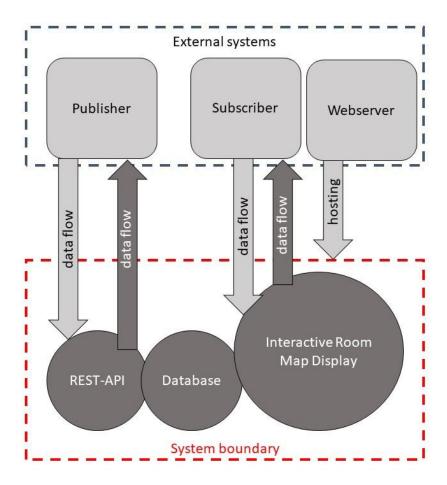
Ein Subscriber hat die Möglichkeit die erhobenen Daten der Publisher über den Web-Service abzurufen und zu abonnieren.

4.6 Webserver

Für das Betreiben des ganzen internen Systems, wird ein Webserver mit entsprechender Infrastruktur benötigt. Die Inbetriebnahme, Verwaltung und Wartung des Webservers sind Sache eines externen Systems.

4.7 Grafik

Es folgt eine Grafik, welche die Systemabgrenzung visualisiert. Die darin enthaltenen Komponenten wurden zuvor erläutert. Die Objekte innerhalb der Systemgrenze stellen das innere System dar. Die Überlappung dieser Objekte repräsentiert eine interne Verbindung.



5 Anforderungen

5.1 Quellen und Vorgehen

5.1.1 Quellen

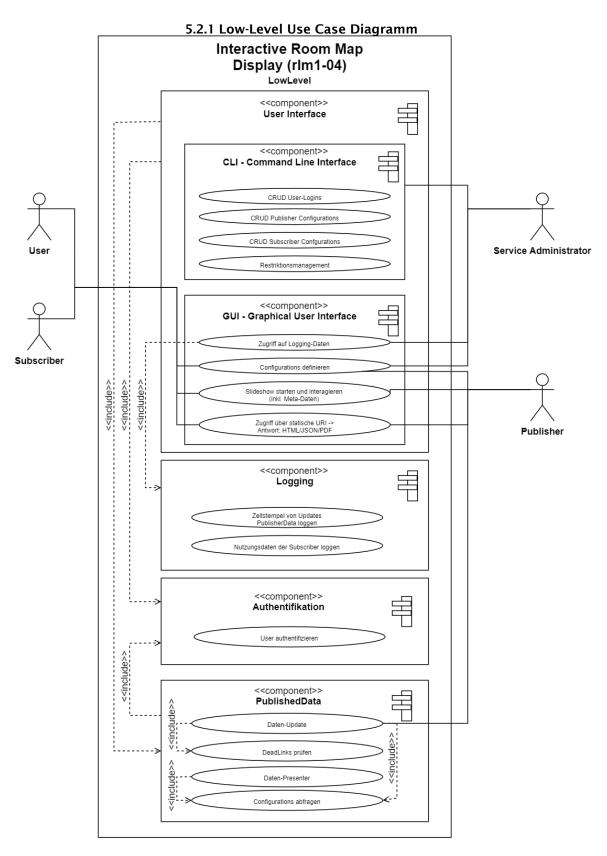
Es folgt eine Liste der verwendeten Quellen für diesen Teil der Arbeit:

- Rolf Dräther, Holger Koscheck, Carsten Sahlin: Scrum kurz & gut (S. 116-118)
- Barbara Ward: Fit für Content Marketing (S. 49-53)
- Thomas Scheuer: Marketing für Dienstleister (S.83)
- Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering (S. 26-34)
- https://moodle.bfh.ch/pluginfile.php/2560528/mod_resource/content/0/rlm1-04-202223-Interactive-Room-Map-Display.pdf
- https://moodle.bfh.ch/pluginfile.php/2560525/mod_resource/content/0/rlm1-01-202223-Room-Management.pdf
- https://moodle.bfh.ch/pluginfile.php/2560526/mod_resource/content/0/rlm1-02-202223-Canteen-Management.pdf
- https://moodle.bfh.ch/pluginfile.php/2560529/mod_resource/content/0/rlm1-05-202223-Indoor-Localisation-Bluetooth.pdf
- https://niceigel.atlassian.net/wiki/spaces/P1/pages/1343489?atlOrigin=eyJpljoiMDJkMGU5YTcxZTFlNGlwY2I4NjlkYjJiZmEwYWEyYzYiLCJwljoiaiJ9
- https://niceigel.atlassian.net/wiki/spaces/P1/pages/3506177?atlOrigin=eyJpljoiNmMxMmFhYj ZmOTE3NDgyY2FiYTcwNzJhNDA1ZTAwNmYiLCJwljoiaiJ9
- https://niceigel.atlassian.net/wiki/spaces/P1/pages/10780673?atlOrigin=eyJpIjoiYjJjZTZIMWU3 YmVhNDI2MThkYmFiZjMyNjEyNGJhNzliLCJwIjoiaiJ9

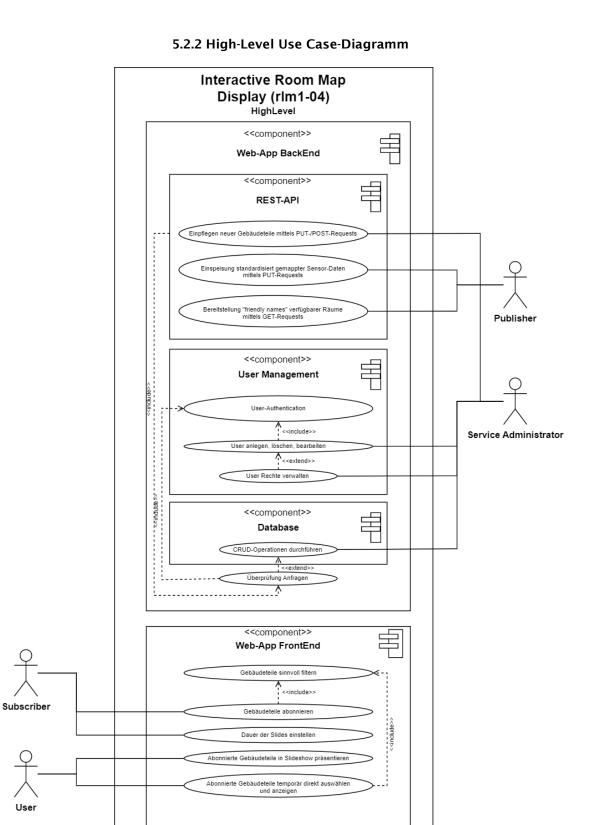
5.1.2 Vorgehen

Nach der ersten Informationsbeschaffung, realisiert über das eigene «Project Proposal», sowie über jener rlm1-04-abhängiger Projekte, wurden in zwei Meetings mit dem PO der Scope des Projekts und erwünschte Use Cases in die Planung der REST-API-Schnittstelle miteinbezogen. Diese wurde als versioniert zu veröffentlichen definiert und war Diskussionsgrundlage für ein erstes Meeting mit sämtlichen verfügbaren rlm1-04-abhängigen Stakeholdern, welches zum Ziel hatte den zuvor diskutierten Scope und die damit einhergehenden Use-Cases zu vervollständigen. Im Kapitel der Anforderungen, in dem Sie sich nun befinden basieren sämtliche Anforderungen auf diesen initialen Absprachen und den darauffolgenden, dem agilen Prozess geschuldeten, Modifikationen dieser initialen Anforderungen, sowie der internen Erkenntnisgewinnung des rlm1-04-Projektteams und allfälliger Kurskorrekturen sämtlicher Stakeholder insbesondere des PO's.

5.2 Funktionale Anforderungen



Auf eine Nummerierung der Use Cases wurde bewusst verzichtet, da eine solche dem agilen Prozess in dieser Skalierung nur bedingt zuträglich wäre.



Auf eine Nummerierung der Use Cases wurde bewusst verzichtet, da eine solche dem agilen Prozess in dieser Skalierung nur bedingt zuträglich wäre.

5.2.3 User Stories (Mindset High Level)

«Als Rolle möchte ich Funktion, um Nutzen/Wert zu erreichen.»

5.2.3.1 User Stories PO (Mindset High Level)

5.2.3.1.1 User Story 1

Als PO (Product Owner) möchte ich eine mit Gebäudeplänen (z.B. SVG) konfigurierbare Map, welche mit Betriebsdaten eines definierten Bereichs gefüttert werden kann, umgesetzt sehen, um den Anforderungen sämtlicher Stakeholder Genüge zu tun.

5.2.3.1.2 User Story 2

Als PO (Product Owner) möchte ich ab der 5. Woche nach Projekt-Beginn die initiale Definition einer REST-API, die den Bereitstellern von Sensor-Daten zur Verfügung steht, um eine reibungslose Zusammenarbeit aller projektabhängiger Entwicklergruppen des Moduls «Project I (BTI3031)» zu gewährleisten.

5.2.3.1.3 User Story 3

Als PO (Product Owner) möchte ich eine browser-basierte Lösung, welche sowohl die Map, als auch Gebäude- und Betriebsdaten ohne manuell getriggerten Reload darstellt, um dem BFH-Gebäude-Nutzer eine Übersicht über die momentan verfügbaren gebäudegebundenen Ressourcen bereitzustellen.

5.2.3.1.4 User Story 4

Als PO (Product Owner) möchte ich dem zukünftigen Entwickler-Team beim Deployment automatisiertes Untit-Tests bereitstellen, um die bestehenden Funktionalitäten (insbesondere REST-API Bezogene) in jedem Fall zu erhalten.

5.2.3.2 User Stories BFH-Gebäude-Admin (Mindset High Level)

5.2.3.2.1 User Story 1

Als BFH-Gebäude-Admin möchte ich die Möglichkeit haben jedes Gebäude und jedes Stockwerk per Kommandozeile (PUT-/POST-Requests) mit einer Bilddatei und einer mitgelieferten Range einzupflegen, um meinen Workflow zu vereinheitlichen und für neue Gebäude-Strukturen agil zu bleiben.

5.2.3.2.2 User Story 2

Als BFH-Gebäude-Admin möchte ich beim Einpflegen neuer Gebäude unabhängig von allfälligen Bereitstellern von Sensor-Daten agieren können, um die klare Trennung der Zuständigkeiten beizubehalten.

5.2.3.2.3 User Story 3

Als BFH-Gebäude-Admin möchte ich mich auf ein Beständiges Daten-Management verlassen können, um mir, und den vom PO bestimmten User-Gruppen alleinige Hoheit über das Gebäude-Management zuzusichern.

5.2.3.3 User Stories BFH-Gebäude-Nutzer (Mindset High Level)

5.2.3.3.1 User Story 1

Als BFH-Gebäude-Nutzer möchte ich eine von mir gewählte Zusammenstellung von Gebäude-Ausschnitten und den dazu aktuell verfügbaren Mess-Daten auf der jeweiligen Map als Slideshow präsentiert bekommen, um mir und den von mir zu informierenden BFH-Gebäude-Besuchern den Aufenthalt effizienter gestalten zu können.

5.2.3.3.2 User Story 2

Als BFH-Gebäude-Nutzer möchte ich meine Zusammenstellung von Gebäude-Ausschnitten und den dazu aktuell verfügbaren Mess-Daten gespeichert haben, dynamisch halten und von Zugriffen ausserhalb geschützt wissen, um die Möglichkeit zu haben als Verantwortlicher meinen Besuchern konstant die zu dem jeweiligen Gebäude passenden und verfügbaren Informationen bereitzustellen.

5.2.3.3.3 User Story 3

Als BFH-Gebäude-Nutzer möchte ich meinen BFH-Gebäude-Besuchern auf lange Sicht (unter Einbezug zukünftiger Entwicklerteams) eine interaktive Oberfläche bieten, die es Ihnen erlaubt in den Verlauf der Slideshow Einfluss zu nehmen, um Ihnen einen Mehrwert nicht durch unnötiges Warten zu verwehren.

5.2.3.4 User Stories BFH-Gebäude-Besucher (Mindset High Level)

5.2.3.4.1 User Story 1

Als BFH-Gebäude-Besucher möchte ich am Eingang eines BFH-Gebäudes innerhalb kurzer Zeit eine Übersicht über die Raumaufteilung der einzelnen Stockwerke erhalten, um meinen Aufenthalt effizienter zu gestalten.

5.2.3.4.2 User Story 2

Als BFH-Gebäude-Besucher möchte ich mich nicht durch komplexe Menüs geführt werden, um zu meinen gewünschten Informationen zu gelangen.

5.2.3.4.3 User Story 3

Als BFH-Gebäude-Besucher möchte ich im gewohnten Corporate Design der BFH bedient werden, um meine Benutzererfahrungen im gewohnten Fluss der BFH beibehalten zu können.

5.2.3.5 User Stories Bereitsteller von Sensor-Daten (Mindset High Level)

5.2.3.5.1 User Story 1

Als Bereitsteller von Sensor-Daten möchte ich ab Woche 5 nach Projekt-Start einen Kern der REST-API definiert wissen, der in seiner Funktionalität zukünftig eher wächst als schrumpft, um meine Daten-Auswertung in standardisierter Form bereitstellen zu können.

5.2.3.5.2 User Story 2

Als Bereitsteller von Sensor-Daten möchte ich die Möglichkeit haben anhand GET-Requests abzufragen, welche Koordinaten zu welchen Räumlichkeiten («friendly names») gehören, um einer ansonsten extensiven Fehlerbehandlung vorzubeugen und einen angenehmeren Workflow während der Bewirtschaftung zu garantieren.

5.2.3.5.3 User Story 3

Als Bereitsteller von Sensor-Daten möchte ich mittels PUT-Requests meine an ein standardisiertes Koordinatensystem gemappten Mess-Daten an die Lösung weitergeben, um meinen internen Projektanforderungen gerecht zu werden.

5.2.3.5.4 User Story 4

Als Bereitsteller von Sensor-Daten möchte ich möglichst bald Zugang zu einer produktiven Umgebung erhalten, um projektgruppen-interne Anforderungen zu verifizieren.

5.2.3.6 User Stories rlm1-04-Entwicklerteam (Mindset High Level)

5.2.3.6.1 User Story 1

Als rlm1-04-Entwicklerteam möchte ich einen realisierbaren Anforderungskatalog mit richtungsweisendem Scope umsetzen, um neben korrekt geführter agiler Prozesse auch ein nachhaltiges Produkt abliefern zu können.

5.2.3.6.2 User Story 2

Als rlm1-04-Entwicklerteam möchte ich anderen Stakeholdern stets Transparenz bieten können, um die Zusammenarbeit zu vereinfachen und die Kommunikationskultur zukünftig beteiligter Stakeholder aufzugleisen.

5.2.3.6.3 User Story 3

Als rlm1-04-Entwicklerteam möchte ich eine browser-basierte Lösung mit REST-API und definiertem Technlologiestack abliefern, um sowohl den technischen und finanziellen Rahmenbedingungen der BFH gerecht zu werden, als auch jedem Endbenutzer einen greifbaren Nutzen zu verschaffen.

5.2.3.7 User Stories Zukünftiges Entwicklerteam (Mindset High Level)

5.2.3.7.1 User Story 1

Als Zukünftiges Entwicklerteam möchte ich auf ein begonnenes Projekt stossen, welches solide aufgesetzt und nach dem Open-Closed-Prinzip gestaltet wurde, um adäquat auf zukünftige Anforderungen reagieren zu können.

5.2.3.7.2 User Story 2

Als Zukünftiges Entwicklerteam möchte ich ein Dokument vorfinden, welches den aktuellen Stand des Projekts und der umgesetzten Anforderungen beinhaltet, um eine rasche Einarbeitung neuer Entwickler zu fördern.

5.2.4 User Stories (Mindset Low Level)

5.2.4.1 User Stories Service Owner (Mindset Low Level)

5.2.4.1.1 User Story 1

Als Service Owner möchte ich ein System, welches einfach zu installieren, konfigurieren, auszuführen und warten ist, um meine anstehenden Arbeiten effizient zu erledigen.

5.2.4.1.2 User Story 2

Als Service Owner möchte ich, dass das Software-System auf bewährten PHP-Komponenten (Frameworks + Libraries) aufgebaut ist, um eine reibungslose Verwaltung mit bestehender Infrastruktur zu gewährleisten.

5.2.4.1.3 User Story 3

Als Service Owner möchte ich Software, die verlässliche Sicherheitskonzepte umsetzt, um eine solide Benutzererfahrung zu gewährleisten. Insbesondere ein adäquates User-/Login-Management, welches garantiert, dass Publisher mit dedizierten Rechten lediglich ihren eigenen erstellten Content überschreiben dürfen und ausgewählte Subscribern das Recht zugeschrieben wird Inhalte auf ihren Screens und Devices darzustellen.

5.2.4.2 User Stories Service Administrator (Mindset Low Level)

5.2.4.2.1 User Story 1

Als Service Administrator möchte ich die Möglichkeit haben sämtliche User-Logins zum Service via Console-Interface aufzuführen, zu erstellen, lesen, updaten und zu löschen, um das Login-Management einfach zu handeln.

5.2.4.2.2 User Story 2

Als Service Administrator möchte ich die Möglichkeit haben sämtliche Publisher-Konfigurationen via Console-Interface aufzuführen, zu erstellen, lesen, updaten und zu löschen, um das Berechtigungs-Management einfach zu handeln.

5.2.4.2.3 User Story 3

Als Service Administrator möchte ich die Möglichkeit haben sämtliche Publisher- und Subscriber-Konfigurationslinks unter Berücksichtigung einer definierten ACL (=Access Control List) via Console-Interface aufzuführen, zu erstellen, lesen, updaten und zu löschen, um das Berechtigungs-Management einfach zu handeln.

5.2.4.2.4 User Story 4

Als Service Administrator möchte ich die Möglichkeit haben sämtliche Subscriber-Konfigurationen via Console-Interface aufzuführen, zu erstellen, lesen, updaten und zu löschen, um das Berechtigungs-Management einfach zu handeln. Auch möchte ich Restriktionen bez. IP-Adress-Bereiche (inkl. Wildcards), und Authentifizierung verhängen zu können, welche optional mit Listung der ID's und Wildcards umgesetzt wurden, um bessere Kontrolle der Zugänge zu gewinnen.

5.2.4.2.5 User Story 5

Als Service Administrator möchte ich Zugriff auf Publisher-Informationen bzgl. veröffentlichter und geupdateter Informationen, sowie auf Protokolle von Nutzungsdaten der einzelnen Subscriber, um den Service-Level sauber dokumentieren zu können und die Rolle als Super-Visor vereinfacht wahrnehmen zu können.

5.2.4.3 User Stories Publisher (Mindset Low Level)

5.2.4.3.1 User Story 1

Als Publisher möchte ich Inhalte unter Berücksichtigung der vom Service Administratoren erstellten Publisher-Konfigurationen und nach Anmeldung via HTTP-PUT-Requests über eine REST-Schnittstelle an den Service senden können, um meine aufbereiteten Daten über einen vereinheitlichten Weg zur Verfügung stellen zu können.

5.2.4.3.2 User Story 2

Als Publisher möchte ich vom Service Rückmeldungen erhalten, ob meine jeweiligen Operationen erfolgreich, oder erfolglos verlaufen sind, um meine aufbereiteten Daten zuverlässig bereitstellen zu können.

5.2.4.3.3 User Story 3

Als Publisher möchte ich über ausreichende Sicherheiten darüber verfügen, dass sowohl Inhalte anderer Publisher vor einer Überschreibung meinerseits gesichert sind, als auch vice-versa, um eine unnötige Fehlersuche zu verhindern.

5.2.4.3.4 User Story 4

Als Publisher möchte ich Zugriff auf ein Low-Level-Interface haben, welches mir erlaubt dem Service pure HTML-Seiten via REST-Schnittstelle zu übergeben, um detailliert spezifizierte Formatierungen und Daten im Service korrekt übersetzt zu wissen.

5.2.4.3.5 User Story 5

Als Publisher möchte ich Zugriff auf Nutzungsinformationen, wie die Anzahl der Aufrufe meiner veröffentlichten Inhalte haben, um die Benefits meiner Arbeit und deren Positionierung besser einschätzen zu können.

5.2.4.4 User Stories Subscriber (Mindset Low Level)

5.2.4.4.1 User Story 1

Als Subscriber möchte ich, abhängig von meinen Subscriber-Konfigurationen, Inhalte von ausgewählten Publishern in meinen Browsern aufrufen und darstellen können, um mein Bedürfnis nach gebäudespezifischen Informationen tilgen zu können.

5.2.4.4.2 User Story 2

Als Subscriber möchte ich Zugang zu Meta-Informationen, wie Zeitpunkt des letzten Updates oder Herkunft über die von abonnierten Publishern veröffentlichten Inhalte haben, um den Ist-Zustand besser einordnen zu können.

5.2.4.4.3 User Story 3

Als Subscriber möchte ich Einfluss auf die Durchlaufgeschwindigkeit der einzelnen Seiten haben, um mein Nutzer-Erlebnis angenehmer zu gestalten.

5.2.4.4.4 User Story 4

Als Subscriber möchte ich die Verfügbarkeit eines Zugriffs auf die von ausgewählten Publishern veröffentlichte Daten per statische URI im Format «HTML» gesichert wissen, um diese in meinem Browser darstellen zu können.

5.2.4.4.5 User Story 4

Als Subscriber möchte ich die Verfügbarkeit eines Zugriffs auf die von ausgewählten Publishern veröffentlichte Daten per statische URI im Format «JSON» gesichert wissen, um diese in Third-Party-Applikationen weiterverarbeiten zu können.

5.2.4.4.6 User Story 5

Als Subscriber möchte ich die Verfügbarkeit eines Zugriffs auf die von ausgewählten Publishern veröffentlichte Daten per statische URI im Format «PDF» gesichert wissen, um diese nach simplem Download in meiner bevorzugten PDF-Applikation in lesbar und druckbarer Form weiterverarbeiten zu können.

5.2.4.1 User Stories User (Mindset Low Level)

5.2.4.1.1 User Story 1

Als User möchte ich von öffentlichen Screens, oder via eigenem Gerät mit relevanten Informationen versorgt werden, um mein BFH-Erlebnis angenehmer zu gestalten.

5.2.4.1.2 User Story 2

Als User möchte ich Meta-Daten zu den an mich herangetragenen Informationen, wie Publisher, Zeitstempel des letzten Updates zusammen auf demselben Kommunikationskanal bereitgestellt wissen, wie die Information selbst, um die Relevanz der mir angezeigten Daten besser einstufen zu können.

5.3 Nicht-funktionale Anforderungen

5.3.1 Reliability

- Keine Auswirkungen fehlerhafter API-Requests auf Slideshow der User
 - o Letzter valider Status wird angezeigt
 - Meta-Daten, wie Timestamps des letzten Uploads werden angezeigt
- Bei Ausfall von Datenfluss Sensorik -> Meldung erstatten
- Möglichkeit dem Nutzer geplante Maintenance-Arbeiten im Vorfeld mitzuteilen
- Während Maintenance Umleitung auf «UnderConstruction-Site»

5.3.2 Ease of use

- Benutzer:
 - o Nachvollziehbarkeit bez. HTTP-Status-Codes bei Verwendung von REST-API
 - Zeit für Rückmeldung fehlerhafter REST-API-Requests <10s
 - o Versionierte Beschreibung der REST-API im .yaml-Format
 - o Komplexitätsreduktion: Nie mehr als 3 Unterkategorien im FrontEnd
 - Fehlercodes in User-Jargon bei FrontEnd
- Entwickler:
 - Verwendeter Technologie-Stack:
 - PHP 7.4.29
 - Laravel 8
 - CSS
 - HTML5
 - PLESK (Hosted Webserver)

5.3.3 Speed

- Page-Refresh in unter 5 Sekunden, bei nicht-erfüllen Kontaktaufnahme PO
- User anlegen für durchschnittlichen Benutzer in inter 10 min möglich inkl. Abonnieren gewünschter Gebäudeteile in Slideshow

5.3.4 Robustness

• (-) -> rlm1-04-Entwicklerteam hat keinen Einfluss auf Robustness Webserver

5.3.5 Portability

- Lauffähig auf jedem gängigen Browser zum Zeitpunkt der Modul-Bewertung
- Sofern sinnvoll: Dynamische Darstellung für verschiedene Auflösungen

5.3.6 Size

- Anzahl User, sowie Umfang Code-Base und Dokumentation nie in Bereich wachsen lassen, der entweder kostenpflichtige Upgrades oder Anfrage an BFH-IT nach sich ziehen würde
 - o z.B. GitLab BFH limitiert auf <= 10GB

5.3.7 Integrity

- Prüfung jeder Anfrage bzgl. Erweiterung REST-API-Funktionalität von Bereitstellern von Sensor-Daten
- Transparente Projektübergabe an PO, um Möglichkeit sicherzustellen von umfänglicher Einweisung zukünftiger Entwicklerteams

5.4 Domänenanforderungen

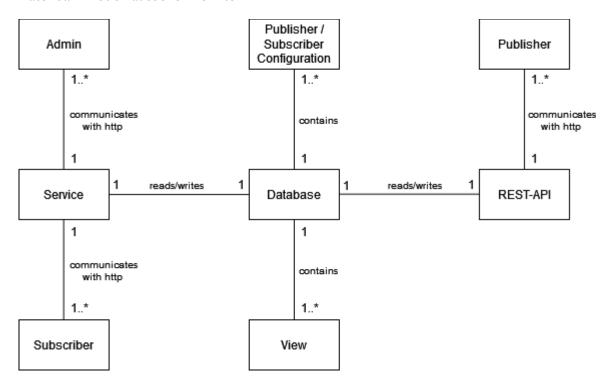
In der Domäne der Web-Applikationen gibt es folgende Anforderungen:

- Die Web-App benötigt einen Server mit einer stabilen Erreichbarkeit
- Die Web-App benötigt einen Server mit einer stabilen Geschwindigkeit
- Die Web-App benötigt einen Server, auf welchem Webservices betrieben werden können.
- Die Web-App benötigt einen Server, auf welchem eine Website betrieben werden kann.
- Die Web-App benötigt einen Server, auf welchem eine Datenbank betrieben werden kann.

6 Domänenmodell

Das Domänenmodell zeigt die Entitäten und dessen Beziehungen untereinander auf. In diesem Falle gibt es die Entitäten Service, Database, REST-API, Admin, Publisher, Subscriber, View und Configuration. Der Service steht hierbei für den Web-Service «Interactive Room Map Display», welcher die eigentliche Web-Applikation darstellt.

Wie diese verbunden sind, sieht man anhand der Linien und deren Beschreibungen. So kann die REST-API mit mehreren Publishern interagieren eine View hingegen nur aus einer Datenbank gelesen werden. Diese Zuteilungen geben auch indirekt Aufschluss über Abläufe oder wie ein Datenbankmodell aussehen könnte.

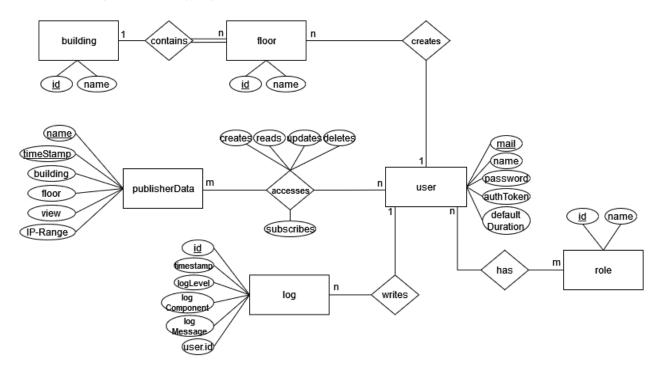


7 Entity Relationship Diagram

Das Entity Relationship Diagram stellt alle involvierten Entitäten und deren Beziehungen untereinander grafisch dar. Dieses Diagramm repräsentiert das Datenbank-Design. Jeder Entitätstyp und jede n-zum-Beziehung stellt eine Tabelle dar, während die jeweiligen Attribute (auch implizit) deren Spalten sind. Im Falle des Interactive Room Map Display Services setzen sich die Entitätstypen aus den «user» deren «role» und «log», den «publisherData», «floor» und deren «building» zusammen.

Ein User greift entsprechend der Berechtigung in der «accesses»-Tabelle auf «publisherData» zu. Ein User mit der Rolle eines Administrators kann «user», «role», «floor» und «building» Objekte bearbeiten bzw. neu erstellen.

Zwecks Logging wird jede Aktion, welche von einem User ausgeht, mit einem Zeitstempel versehen und in der «log»-Tabelle abgelegt.



8 DB Sicherheit

Als Teil der Anforderungen gewünscht, beschreiben wir hier, wie wir die Datenbank absichern, die Daten darin schützen, sowie die Datenkonsistenz und Sicherung der Daten.

8.1 Zugriffskontrolle

Bei der Zugriffskontrolle soll das Prinzip der Rollentrennung eingesetzt werden. Es wird für jede Rolle, einen eigenen Datenbankbenutzer geben.

Konkret also, wird es die folgenden Benutzer geben:

- Service
- Service Administrator
- Publisher room
- Publisher canteen
- Publisher object
- Subscriber

Bei den Benutzern wird jeweils das Prinzip der minimalen benötigten Berechtigungen durchgesetzt. Die Berechtigungen werden, wie in Kapitel 8 beschrieben, gesetzt.

Die Rechteverteilung wird durch den Service Owner vorgenommen, wobei dies initial aus dem Projektteam und den Migration-skripten kommen wird.

Die Verifikation der Rechte wird durch Unittests sichergestellt.

8.2 Authentifizierung

Die Authentisierung an der Datenbank erfolgt via Passwort. Das Reglement des Passworts ist folgendes:

- Mind. 20 Zeichen
- Pflicht 1 Sonderzeichen
- Pflicht 1 Zahl
- Pflicht 1 grossgeschriebener Buchstabe
- Pflicht 1 kleingeschriebener Buchstabe

Um zusätzliche Sicherheit zu bieten, werden die Zugänge eingeschränkt via IP-Ranges. Des Weiteren werden nur bewährte Standardtechnologien eingesetzt, welche in der Praxis vielfach erprobt sind, um weiteren Problemen vorzugreifen.

8.3 Verschlüsselung

Die Verschlüsselung der Datenübertragung wird via SSL gelöst.

8.4 Konsistenz

Um die Datenkonsistenz sicherzustellen, werden grundsätzlich alle Datenbankänderungen via Migrationsskripte über das Laravel-Framework angestossen. Des Weiteren werden Queries nicht direkt ausgeführt, sondern via beginTransaction()-Funktion von Laravel angestossen und mit der commit()-Funktion ausgeführt, oder bei Fehlern via rollback()-Funktion abgebrochen. Hier wird auch pessimistic Locking eingesetzt, via sharedLock()-Funktion oder lockForUpdate()-Funktion von Laravel.

8.5 Datensicherung

Die Datensicherung wird täglich um 00:00 via Laravel artisan backup angestossen und lokal auf dem Filesystem abgelegt.

9 ACL Konzept

DB Tables
R=read
W=write
U=update
D=delete
Pub_data = Published Data
Sub_Config = Subscriber Config

Rolle/ACL																												
	r User	w User	u User	d user	R floor	W floor	U floor	D floor	R building	W building	U building	D building	R pub_data	W pub_data	U pub_data	D pub_data	R sub_config	W sub_config	U sub_config	D sub_config	R log_data	W log_data	U log_data	D log_data	R backup	W backup	U backup	D backup
Service	Х	Х	Х	Х	х	х	х	Х	Х	Х	х	х	Х	х	Х	х	Х	х	Х	х	Х	Х	х	Х	Х	Х	Х	х
Service Owner	х	х	х	х	х	х	х	х	х	х	х	х	х			х	х	х	х	х	х				Х	Х		X
Service	Х	Х	х	х	х	х	х	х	х	х	Х	Х	х			Х	Х	Х	Х	Х	Х				Х	Х		X
Administrator																												
Publisher													Х	Х							Х							
roomdata																												
Publisher Mensa													Х	Х							Х							
Publisher Object													Х	Х							х							
Subscriber													х				х		Х									
User BFH intern													Х															
User Public													х															

10 Sicherheitskonzept REST-API

10.1 Generell

- Die REST API soll auf folgender URI abrufbar sein: <a href="https://service.p1-irm.bfh-web-labs.ch/api/<version>/...">https://service.p1-irm.bfh-web-labs.ch/api/<version>/...
- Die Versionierung wird im URI-Pfad realisiert, z. B. https://service.p1-irm.bfh-web-labs.ch/api/v1/...
- Alle Versionen mit wesentlichen Erweiterungen/Änderungen erhalten ihren eigenen v... Code

10.2 Authentifizierungsmethode

Gemäss Vorgabe wird die Authentifizierung mit «Authentication Tokens» abgehandelt. Für die Implementierung wird das «Sanctum» Paket von Laravel verwendet.

Nach erfolgreichem Login (Basic Authentication) wird in der http-Response ein AuthToken an den Client geschickt, welcher bei Anfragen auf geschützte Routes im «Authorization»-Header mit Schema «Bearer» mitgeschickt werden muss. Der Token ist für 24 Stunden gültig, ehe durch erneute Anmeldung ein neues gelöst werden muss.

10.3 Verschlüsselung

Standardmässig wird die Verbindung mit TLS (Transport Layer Security) verschlüsselt. Diese ist Bestandteil der Serverinfrastruktur, auf welcher die REST-API betrieben wird.

10.4 Quotas and Throttling

Um sicherzustellen, dass nicht zu viele Aufrufe in einer bestimmten Zeit auf einer Route erfolgen, wird ein «Rate Limiter» eingesetzt. Dieser wird mithilfe der «RateLimiter» Klasse von Laravel implementiert. Um die API vor Spikes und Denial-of-Service-Attacken zu schützen, werden Throttling-Regeln erstellt. Dafür wird die von Laravel zur Verfügung gestellte Middleware «Throttle» verwendet. Generell gilt eine Limite von 60 Requests pro Minute, ausgehend vom selben Client. Beim Login sind es 10 pro Minute.

10.5 Contenttype Validation

Standardmässig wird nur JSON-Format akzeptiert.

10.6 Input Validation

Um das erhaltene HTML zu validieren, wird ein HTML Sanitizer eingesetzt. Dafür wird die PHP Library «HTML Purifier» verwendet. Gemäss Vorgaben werden grundsätzlich alle Tags, welche externe Quellen referenzieren, gelöscht. Images werden nur eingebettet in Form von Base64 akzeptiert. Des Weiteren wird auch eingebettetes JavaScript gelöscht.

10.7 Error Handling

Jede Anfrage wird mit einem entsprechenden Statuscode beantwortet:

Use Case	HTTP Status Code
Invalid Accept Header	400 Bad Request
Unauthenticated / bad credentials	401 Unauthorized
User not allowed to access resource	403 Forbidden
Resource not found	404 Resource not found
Method Not Allowed	405 Method Not Allowed
Unsupported Media Type	415 Unsupported Media Type
Bad formed request body	422 Unprocessable Entity
Too Many Requests	429 Too Many Requests

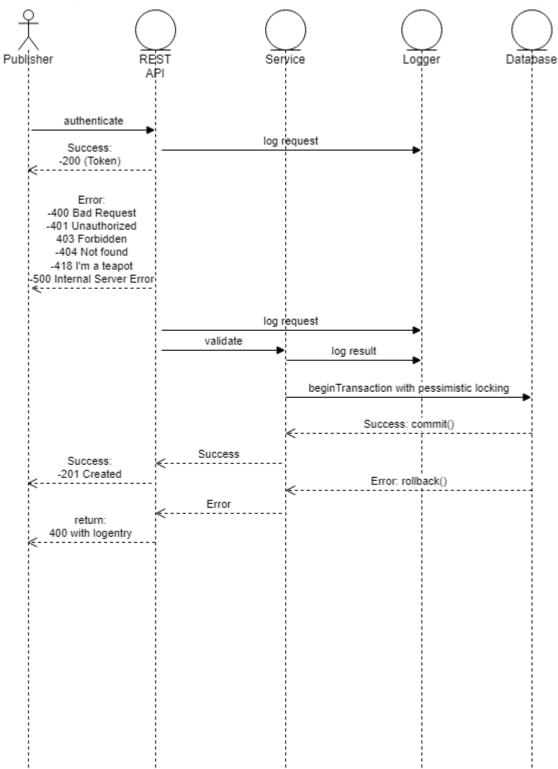
Somit werden keine unnötigen detaillierten Angaben an den Client gebracht, welche für einen Angriff nützlich sein könnten.

10.8 Response Headers

Header	Begründung
Cache-Control: no-store	Verhindert, dass sensitive Daten im Cache
	abgespeichert werden können.
Content-Security-Policy: frame-ancestors 'none'	Schützt gegen drag-and-drop style clickjacking
	Attacken
Content-Type: application/json	Um den Content-Type zu spezifizieren
Strict-Transport-Security: 'max-age=31536000'	Erlaubt nur HTTPS Verbindungen und schützt
	vor gespooften Zertifikaten
X-Content-Type-Options: nosniff	Um zu verhindern, dass Browser MIME-Sniffing
	betreiben und Antworten unangemessen als
	HTML interpretieren.
X-Frame-Options: DENY	Schützt gegen drag-and-drop style clickjacking
	Attacken

11 Ablauf Publishing-Prozess

Der Publishing-Prozess und damit die Kommunikation mit den Stakeholdern, wird wie folgt ablaufen:



12Literaturverzeichnis

12.1 Quellen betreffend (Nicht-) Funktionalen Anforderungen

12.1.1 Literatur FA/NFA

Rolf Dräther, Holger Koscheck, Carsten Sahlin: Scrum - kurz & gut, 978-3-96009-094-6

Barbara Ward: Fit für Content Marketing, 978-3-86980-310-4

Thomas Scheuer: Marketing für Dienstleister, 978-3-640-50269-1

Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering, 978-3-86490-283-3

12.2 Quellen zu Sicherheitskonzept REST-API

Laravel Documentation: Sanctum:

URL: https://laravel.com/docs/8.x/sanctum#api-token-authentication (Stand: 01.11.2022).

Laravel Documentation: Routing:

URL: https://laravel.com/docs/9.x/routing#rate-limiting (Stand: 01.11.2022).

OWASP: HTML Sanitizer:

URL: https://owasp.org/www-project-java-html-sanitizer/ (Stand: 01.11.2022).

OWASP Cheatsheets: Security Headers REST: URL:

https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html#security-headers

(Stand: 01.11.2022).

13 Anhang

13.1 Anhänge bzgl. Anforderungen

13.1.1 Project Proposal rlm1-04

BTI3031 - Project 1 Autumn 2022-23

ID: rlm1-202223-04

Title: Interactive Room Map Display (by API)

Context

For building management purposes, interactive maps are a very important element. Typical applications include building occupancy, efficient use of space, temperature or crowd control. Interactive maps are based on a customised map of a floor or building, add operational data such as temperature, position of tracked objects, or occupancy, by using colors (heat maps), gauges, or signs for alert.

Use Cases, Requirements

The objective of the project is to:

- define a configurable map, which can be configured with floor plans (SVG, as provided on https://bfh-pp.pocketcampus.org/map/search) and fed with operational data of a definable range of kinds
- design a REST API (interface available by semester week 5, latest), available to other groups in "Project 1" (rlm1-01, rlm1-02 and rlm1-05)
- implement a browser-based solution, capable to display the map with base and operational data, without manual reload.

Technology

The solution shall use web technologies and be hosted on the https://current-info.bfh.science platform (BFH infrastructure container, PHP 7.4, MySQL).

If PHP/MySQL is used, then the Laravel Framework is clearly preferred (https://laravel.com).

Supervisor

Michael Röthlin - rlm1

Michael Röthlin

13.1.2 Project Proposal rlm1-01

BTI3031 - Project 1 Autumn 2022-23

ID: rlm1-202223-01

Title: Room Management Solution for SIPBB

Context

In the Switzerland Innovation Park Biel-Bienne (SIPBB), rooms are sparse and have to be managed carefully. Group work rooms may be reserved in advance (but not being used nevertheless) or not being reserved but used in an ad-hoc manner by groups of students. In addition, the presence (availability) of support staff at the reception desk S244 is an important information.

In this project, wall mounted occupancy sensors shall be used to supervise both the availability of group work rooms and the reception desk. The information of the sensors is collected by IoT Gateway, processed and dispatched to client devices.

Use Cases, Requirements

The solution to be explored here is based on Enocean wireless technology (http://enocean.com). The occupancy sensors employed (https://www.enocean.com/en/product/wall-mounted-occupancysensor-eosw-oem/?frequency=868) are energy harvesting devices, removing the need for batteries.

At the end of the project, availability of meeting rooms and occupancy of the reception room are to be displayed on maps, served by the https://current-info.bfh.science platform (BFH infrastructure container), using the API provided by group rlm1-04.

Technology

Basis of the solution is the Jeedom Atlas Home Automation Gateway (https://www.jeedom.com/en/box.html).

Therefore, Web technologies (PHP, MySQL, JavaScript) shall be explored and applied for production purposes.

Supervisor

Michael Röthlin - rlm1

Michael Röthlin

13.1.3 Project Proposal rlm1-02

BTI3031 – Project 1 Autumn 2022-23

ID: rlm1-202223-02

Title: IoT in the gastronomy sector at BFH Rolex Building

Context

The Faculty of Computer Science disposes of an already installed, but as yet hardly used IoT gateway. This device allows signals to be received and transmitted in the range of 868 MHz (Enocean standard), and the wireless signals can then be forwarded via Ethernet or processed on the device itself (integrated Node-RED editor).

Use Cases, Requirements

The objective of the project is to:

- fully intercept already installed (but not yet used) sensors on opening/closing of doors or presence/movement of persons, to store the collected data in a database
- generate visualisations of the frequency of use, e.g. of our canteen or the canteen terrace
 (https://www.bfh.ch/de/ueber-die-bfh/standorte-infrastruktur/standorte/biel-hoeheweg/), based on the collected data
 and to make them available on an intranet website. Of course, the placement of the sensors must be checked and
 optimised if necessary
- Implementing further ideas that arise from dealing with technology in our environment.

At the end of the project, frequency and waiting data are to be displayed in a customer-oriented form on the https://current-info.bfh.science platform (BFH infrastructure container), using the API provided by group rlm1-04.

Technology

Basis of the solution is the Thermokon STC loTG a teway (https://www.thermokon.de/direct/categories/stciot).

Programming can be made on the device (Node-RED) or in a dedicated application. As the https://current-info.bfh.science platform supports only specific Web technologies (PHP, MySQL, JavaScript), these technologies should be considered. If PHP/MySQL, then the Laravel Framework would be preferred (https://laravel.com).

Supervisor

Michael Röthlin - rlm1

Michael Röthlin

13.1.4 Project Proposal rlm1-05

BTI3031 - Project 1 Autumn 2022-23

ID: rlm1-202223-05

Title: Indoor Localization with Bluetooth Technologies

Context

In this project, the current state of the art of localization of objects in a workplace environment shall be explored.

Use Cases, Requirements

The objective of the project is to:

- understand, setup and operate the Mikro TikKNOT IoT Gatewayrouter (https://mikrotik.com/product/knot); with its Bluetooth interface, the router can be used for asset tracking and telemetry based on Bluetooth advertisement packets (https://help.mikrotik.com/docs/display/ROS/Bluetooth)
- integrate standard industry Bluetooth beacons to track assets in the office space (at SIPBB)
- to process and forward tracking information about assets on an interactive map via its REST API (to be provided by other Project 1 group rlm1-04), to integrate it on the https://currentinfo.bfh.science.platform (BFH infrastructure container)
- · evaluate the robustness and stability of the overall solution
- implement further ideas that arise from dealing with technology in the environment.

At the end of the project, frequency and waiting data are to be displayed in a customer-oriented form.

Technology

- Basis of the solution is the MikroTik KNOT IoT Gateway RB924i-2nD-BT5&BG77 router (https://mikrotik.com/product/knot)
- TELTONIKA BC ID Bluetooth Beacon and
- TELTONIKA BS ID Bluetooth Beacon.

Application programming can be made either on the router device/Router OS (https://help.mikrotik.com/docs/display/ROS/RouterOS) or in a dedicated application.

As the https://current-info.bfh.science platform supports only specific Web technologies (PHP, MySQL, JavaScript), these technologies should be considered. If PHP/MySQL, then the Laravel Framework would be preferred (https://laravel.com).

Supervisor

Michael Röthlin - rlm1

Michael Röthlin1

13.1.5 2022-09-27 Meeting mit Herr Röthlin

Datum

27. Sept2022

▼ Teilnehmer

- @Nicolas Waser
- @Marc Rudolf Fuhrer
- @Nic Eigel
- @Michael Röthlin

Ziele

- Briefing
- Fragen aus vorheriger Sitzung behandeln

Diskussionsthemen

Eintrag	Notizen	
Vision	 Map auf iPad oder Bildschirm im BFH Campus Map dynamisch: Serie aus wenigen wechselnden Bildschirmen (z. B. all 5 Sekunden wechseln), evt. sogar Live Tracking 	
Anforderungen	 REST API Map auf Website darstellen prinzipiell: unabhängig von SIPBB (alle BFH Räumlichkeiten) automatisierte Daten Einspeisung (mit Unit Tests?) 	
Optional	 Admin Seite (Webbasierte konfiguration) Historie (z. B. FlipChart rückverfolgen) Live Tracking Smartphone App 	
Eingespiesene Daten	Kantine Wartezeit freie/besetzte Plätze Räume Belegungsinformation (Aktivität/Status) Gruppenraum frei/besetzt Frau Locher anwesend/abwesend Indoor Lokalisierung wo steht FlipChart (Raum oder Koordinaten) etc.	
2DO	 Besprechung 4.10.22: Einladung an Herr Röthlin senden Backlog Stories erstellen GIT Repo erstellen (Herr Röthlin einladen) genauer Umfang des Datenaustausches an Schnittstelle abklären (mit Bild oder ohne) 	
SCRUM	 Product Owner: Michael Röthlin (2 Wochen Meetings) Ticket/Issue-Basierte Sprint-Meetings Master: Nic 	
Technologie	PHP 7.4 Laravel 8.0 Deployment Platform: bfh-web-labs.ch	

13.1.6 2022-10-04 Meeting mit Herr Röthlin

Datum

4. Okt. 2022

▼ Teilnehmer

- @Nicolas Waser
- @Marc Rudolf Fuhrer
- @Nic Eigel
- @Michael Röthlin

Ziele

* API Fragen klären

Diskussionsthemen

Eintrag	Notizen	
Demo Deployment Platform	bfh-web-labs.ch:8443; laravel/php setup	
REST API	mit oder ohne Bilddatei uns überlassen	
Admin Panel	nicht über API	
Navigation	z. B. SIPBB 2. Stock	
automatische Änderung	 REST-API automatisch Änderungen auf Karte? (nur wenn Änderung) dynamische Grafik oder alle paar Sekunden von DB aktualisieren 	
bewegende Objekte	letzte Position?	
Definition nächster Schritte	 Epics definieren Backlog füllen Sprint definieren erster Sprint starten 	

13.1.7 2022-10-18 Besprechungsnotizen (Abstimmungsmeeting projektabhängiger Stakeholder)

Datum

18 Okt. 2022

▼ Teilnehmer

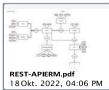
- Hosts: rlm1-04 Interactive Room Map Display (by API)
 - @Nic Eigel (präsent)
 - nicolasacha.eigel@students.bfh.ch
 - @Nicolas Waser (präsent)
 - nicolasanthony.waser@students.bfh.ch
 - @Marc Rudolf Fuhrer(präsent)
 - marcrudolf.fuhrer@students.bfh.ch

Folgende Teilnehmer wurden eingeladen und sind entweder präsent oder werden durch Vertretung über Inhalte informiert:

- rlm1-01 Room Management Solution for SIPBB
 - o Thossy, Michael, thosm1 (präsent)
 - michael.thossy@students.bfh.ch
 - Moser, Florian, mosef5 (präsent)
 - florian.moser@students.bfh.ch
 - Lupberger, Yanis, lupby1 (präsent)
 - yanisjoel.lupberger@students.bfh.ch
- rlm1-02 IoT in the gastronomy sector at BFH Rolex Building
 - o Fabio Bertagna, bertf3 (präsent)
 - fabio.bertagna@students.bfh.ch
 - Yannik Dällenbach, dally1 (präsent)
 - yannik.daellenbach@students.bfh.ch
 - Kunz, Philip, kunzp9 (präsent)
 - philipjonah.kunz@students.bfh.ch
- rlm1-05 Indoor Localization with Bluetooth Technologies
 - Zingg, Nicole, zingn1 (präsent)
 - nicole.zingg@students.bfh.ch
 - Pfeiffer, Thierry, pfeit2 (präsent)
 - thierry.pfeiffer@students.bfh.ch
 - Matthias, Santschi, santm6
 - matthiaslukas.santschi@students.bfh.ch
- Product Owner
 - @Michael Röthlin (präsent)
 - michael.roethlin@bfh.ch

Material







♣ Diskussionsthemen

Zeit	Eintrag	Notizen
18:10	Begrüssung	 Sicherstellen, dass jede Gruppe vertreten und bereitgestellte Dokumente im Vorfeld und zum jetzigen Zeitpunkt Verantwortlichkeiten klären Dreh- und Angelpunkt bei interdisziplinären Fragen und Zuständigkeiten → @Michael Röthlin syaml API Demonstration Begrenzung auf Kern von 4 PUT-Requests Zu verstehen als Entwurf Demonstration Unterteilung Floors anhand: 2. Obergeschoss Beteinste Jahrbeit Gruppen Gruppen Gerichten Gerich
18:17	Verständnisfragen	Nachfrage von @Michael Röthlin Einbeziehung der Mensa Ist das ein Tag oder eine Funktion, oder? Antwort: API-Spezifikation definiert Unterteilung und Funktionalität Ist möglich, dass ein Zusammenschluss von Feldern einen einzigen Raum definiert? Antwort: Ja, wird mit Delimiter ":" realisiert. Wurde in Spezifikation ausgehändigt. Wird der Status automatisch upgedatet? Antwort: Der Status wird von den einzelnen Gruppen geupdatet. Es werden ausschliesslich die PUT-Requests realisiert

18:22	Verantwortlichkeiten Content	Diskussion angestossen von Zingg/Moser/Kunz bzgl. Verantwortlichkeiten der Erfassung von Räumlichkeiten
		Erfassung von Räumlichkeiten Wichtigkeit der Definition der Scopes tritt in Vordergrund. Vor allem weitsichtig überlegt soll auch eine weiterführende Nutzung möglich sein und nicht von unsere Modul-Gruppen abhängig sein. Delegation wird an @Michael Röthlin übergeben @Nic Eigel schlägt vor eine initiale Erfassung durch uns zu gewährleisten, jedoch weiterführung an BFH zu überlassen. @Michael Röthlin Übersicht von Räume/Inventar sollte per GET-Request abrufbar sein. @Nic Eigel verabschiedet sich wegen begrenzter Zeit @Nicolas Waser übernimmt Vertretung rlm1-04 @Michael Röthlin fasst aus User-Sicht Möglichkeiten der API zusammen. Bemängelt, dass kein POST-Request verfügbar ist Initialer Zustand herstellen könnte so schwierig sein Moser denkt an, dass Verantwortlichkeit von Zuweisung von Koordinaten an Gruppe rlm1-04 zu delegieren ist. @Michael Röthlin Mapping von "Quadrätchen" sieht Verantwortung bei einzelnen Gruppen liegen Kunz sieht bei Mensa Schwierigkeiten bei eigener Gruppen, da zu dynamisch. Wünschenswert wäre ein Array von Koordinaten schicken zu können, als einen zu langen String zu interpretieren. Bertagna Mapping von Koordinaten sei komplett über dem Scope der einzelnen Gruppen @Michael Röthlin erläutert Problematik und benennt seine Sicht: Modulares Aufbereiten sollte durch allgemeine API gewährleistet werden. Soll mit Curl über Kommandozeile inkrementell auffüllbar sein. Möglichkeiten von "Friendly-Names" usw. @Marc Rudolf Fuhrer Nennt Problematik, von
		weiterführung des Projekts, falls weitere Sensorik dazukommt. Zeigt Entscheidung auf zwischen Allgemeine API, sodass jede Gruppe in eigenem Tempo arbeitet und so offen für Extensions ist. Spezielle API die z.B. in 2-Wochen-Zyklus mit einzelnen Gruppen erarbeitet wird. Kunz und @Nic Eigel diskutieren Verantwortlichkeiten. Lupberger nennt Error-Handling z.B. von Überlappungen von Räumen als Problem @Michael Röthlin Low-Level, die einzelne Quadrätchen definiert. Higher-Level, die mit
18:54	Weiterführung Verantwortlichkeiten	Abmessungen/Koordinaten abgefüllt wird. • Weg und Format der API inkl. Gedankengänge Semantik ist der Richtige • Dass einzelne Gruppen Quadrätchen zu Räumen zuweisen können. • Einheitliche Schnittstelle für alle Gruppen ist anzustreben • Moser, Zingg: Zuweisung Pixel → Raum ist schon erforderlich • @Nicolas Waser • Zingg: Statisches abfüllen ist nicht praktisch • Moser: Feedback mit GET-Requests wäre erforderlich
19:00	Darstellung Displays + Lösung Verantwortlichkeiten	Diskussion mehrerer Beteiligter mit Unterbreitung Vorschlag angestrebter Vorgehensweise rlm1-04 durch @Marc Rudolf Fuhrer Konklusion: Beispiel alle Display-Standorte sind User, die Möglichkeiten haben sollen: Gebäude/Floors zu abonnieren (praktisches Filtering mit Departementen, Gebäude könnte man andenken) In "Präsentationsmodus" zu wechseln, der Einstellungen beinhaltet, ob: Einzelner Floor angezeigt wird, oder Slideshow mit dynamischem Zeitintervall Konklusion: Angebot von rlm1-04 dem User eine Grafik zurückzugeben mit generiertem Bild, welches jeweils Floor-Grafik mit Unterteilungen und Zonen-Bezeichnern beinhaltet.

19:08 Weiteres Vorgehen	 Es stellt sich heraus, dass rlm1-04 Dreh- und Angelpunkt sämtlicher Gruppen sein wird (Analog zu technischer Abhängigkeit bzgl. REST-API). Verantwortlichkeit wird wahrgenommen und umgesetzt (siehe Handlungspunkte). Versionierung wird verfolgt und wiefolgt umgesetzt: API-Funktionalität soll beständig bleiben und wird ausschliesslich um Funktionalitäten erweitert, nur in Notfällen dezimiert. Produktives System wird als Stable-Releases verfügbar gemacht und immer über dieselbe URL anzusprechen sein. Stable-Releases werden von rlm1-04 als inkrementelle Releases zu Sprint-Enden zusammen mit kurzen Release-Notes veröffentlicht und per OneNote den Gruppenverantwortlichen zugetragen. Frühere Releases werden nicht mehr verfügbar sein.
-------------------------	--

Handlungspunkte

	Jede Gruppe meldet Kontaktdaten Aeines Verantwortlichen an Onic Eigel . Dieser wird als einziger bei der Kommunikation berücksichtigt.
	rlm1-04 stellt dieses Protokoll den Gruppen-Verantwortlichen inkl. PO @Michael Röthlin zur Verfügung.
	rlm1-04 stellt oneNote-Ordner zur Verfügung, welcher laufend Release-Notes zum letzten Stable-Release mit Begründung enthält.
-	rlm1-04 stellt sobald Implementation es zulässt und Änderungen der API erfolgt sind, zum eweils nächsten Sprint-Ende eine produktive Umgebung zur Verfügung, die jeweils den letzten
	Stable-Release enthält.

Entscheidungen

- Y API wird akzeptiert und zukünftig um Funktionalitäten
- 1 API soll Ihren allgemeinen Charakter beibehalten, um Erweiterungen punkto Sensorik und Gebäude-Management zulassen zu können

13.1.1 2022-10-25 Meeting mit Herr Röthlin



25.10.2022



- Nic Eigel
- @ Marc Rudolf Fuhrer
- Nicolas Waser
- @ Michael Röthlin



Verständnis Mail vom 24.10.2022 um 09:05 Uhr "Fwd: [Confluence] Marc Rudolf Fuhrer has assigned tasks to you in "2022-10-18 Besprechungsnotizen (Abstimmungsmeeting projektabhängiger Stakeholder)""

Material



PDiskussionsthemen

Zeit	Eintrag	Notizen
14:35 (ohne @ Michael Röthlin)	Verständnis Mail	 Klärung Punkt 3 von "Low Level API Requirements irm.pdf" Siehe UseCase-Diagramm (User Management)
14:48 (ohne @ Michael Röthlin)	• "	 Klärung overall von "Low Level API Requirements irm.pdf Rede war von "abonnierten Stockweren" pro gleichen Display Bedarf Ausführungen, wieso Publishers HTML mit Daten senden und wir Daten auf Grafik zeichnen.
15:00	Erklärung Mail von	Zweck war definierte Anforderungen seitens PO.
15:01	Heutiges Thema aus Sicht von rlm1-04	Verständnisfragen
15:05	Info seitens Modul-Leitung	Checkliste von Marcel Pfahrer wird noch erscheinen. "README.md" soll alle Links enthalten zu: UseCase-Diagramm Context-Diagramm usw (Mit unserer Gesamtspezifikation sicher über Anforderung)
15:06	Core Besprechung	 PO meint Termin Woche 5 war rlm1-04 unter Druck gesetzt. Darum sehr zufrieden, dass überhaupt wie gewünscht umgesetzt worden. Idee ist Entlastung durch Low-Level-API Keine lokalen Referenzen auf Dokumente, die es nicht mehr gibt, darum HTML Publisher-seitig. HTML checken, ob darstellbar () Wenn check gut (inkl. User-Rechte) per GET-Request gesamter Content abrufbar Schrittweises erarbeiten von gemeinsamen Stylesheets, Fonts, und sich darüber einem gemeinsam
15:18	Kommunikation rlm1-04 abhängiger Gruppen	 Wurde bis anhin noch nicht kommuniziert. PO delegiert Kommunikation an rlm1-04, im Stil von "Stakeholder hat vorgeschlagen () und wurde akzeptiert." oder "Stakeholder stellt neue Anforderungen und geschuldet agilem Prozess: so" Vorteil z.B. von Kantine-Gruppe: Obwohl Aufwand Karte zu zeichnen Sie haben Freiheiten nicht nur auf Gebäudeplänen zeichnen zu müssen, sondern können Ihre bisherige 3x3 Tabelle mit anders gearteten Informationen abfüllen Andere Gruppen wären zum jetzigen Zeitpunkt noch nicht bereit Inhalte in von unserer angedenkter Weise

15:24	• Zsfg.	REST-API User-Authentifizierung Check, ob deadlinks im HTML Darstellen, wenn ok	
15:26	• Frage von rlm1-04	 Ist das eine URL, die DiaShow darstellt, oder sollte das User-gebunden sein? Content-abhängig Gibt Geräte mit Eingabegeräten zu Authentifizierung Gibt Geräte mit IP-Adressen Freigabe URL kann mit "+"-Zeichen und GET-Parametern modifiziert werden 	
15:29	Frage von rlm1-04	Möglichkeit zur Verwaltung von Rollen und Rechten: Soll das über WebApplikation durchführbar sein? Nein, das macht Service-Admin über Kommandozeile. Was sein kann: Das man sich halt anmelden muss. Public-/Private-Keys möglich? Ja! Ausser Ausnahme Log-File (Punkt 5) Für alle authentisierte User (Usr/Pwd) (Von uns zu bestimmen, ob zusammengefasst, oder Einzelpersonen) HTTP-Responses, Nutzungsinformationen, genaueres Fehler-Tracing ermöglichen.	
15:36	Bewertung Konzept	 Publisher = Gutes Konzept Idee des Subscribers soll Projektarbeit in Richtung Sicherheitskonzept pushen (lieber low-level als high-level) 	
15:37	Wen kann rlm1-04 punkto Sicherheit	Herr Benoist steht zur Verfügung	
15:44	Erläuterung Service Owner	Zum Teil Banalitäten vorgegeben aber soll als Absicherung fungieren.	
15:46	API-Versionierung	Group-Befehle in Laravel sind praktisch für mehrere routes, wenn man URL mit "/api/v1" gestalten würde. Abfolge definieren, z.B.: Zuerst GET an "/api/v1/login" senden, um token zu erhalten dann PUT an "/api/v1/" mit diesem token senden ann Bei jeder neuen Version Bitte Update an andere Gruppen senden.	
15:52	Webserver	 Port 80 muss offen sein, da wegen Let'sEncrypt alle Endpunkte erreichbar sein sollen. Bitte Logging befolgen, nicht dass z.B. ganze DB des Webservers unbemerkt hopps geht. 	
15:53	Testing	 Keine externen Tools wie Postman etc. Laravel-interne Tests, Guzzle,verwenden. Laravel-Seeds für DB-Mockdaten verwenden 	
15:58	PLESK/API	Geschützte Verzeichnisse erstellen SwaggerUl anbieten	
16:00	Publisher Requirements	 Logs ist ein Balance-Akt punkto Privacy, aber ist sehr nützlich (z.B. Wie oft wurde meine letzte Information auch wirklich verwendet?,) 	
16:03	Punkte 6+7 von Subscriber	Nice-to-have für in Backlog	
16:08	• Conclusion	 Fokus-Shift auf Sicherheit / API in Tiefe ist sehr interessant und scheint lehrreich Jetziger Sprint wird regulär abgeschlossen Neue Tasks mit höchster Priorität bei nächstem Sprint-Planning Möglichst zeitnah andere Gruppen informieren 	
• Weiteres Voraehen Termine)		n (inkl. 01.11.2022 von 18:00-19:00 (verschobenes Weeklv) 27.11.2022 zwischen 19:00-21:00 Tasks erfassen für P1 Sprint 3 (Nic Figel & Nicolas Waser) Anforderungskatalog nach P1 Sprint 2 wird als High-Level- Anforderungskatalog in Gesamtspezifikation beibehalten Anforderungskatalog Low-Level wird analog zu High-Level im nächsten Sprint der Gesamtspezifikation hinzugefügt Prüfung, ob eigenes Kapitel für Nachvollziehbarkeit Anforderungs-Wechstergänzungen sinnvoll und ob standardisiertes Naming in Scrum existiert	

13.1.1 2022-11-22 Besprechungsnotizen (Abstimmungsmeeting projektabhängiger Stakeholder)



22.11.2022



- Hosts: rlm1-04 Interactive Room Map Display (by API)
 - Wic Eigel (präsent)
 - nicolasacha.eigel@students.bfh.ch
 - @Nicolas Waser (präsent)
 - nicolasanthony.waser@students.bfh.ch
 - @Marc Rudolf Fuhrer (präsent) marcrudolf.fuhrer@students.bfh.ch

Folgende Teilnehmer wurden eingeladen und sind entweder präsent oder werden durch Vertretung über Inhalte informiert:

- rlm1-01 Room Management Solution for SIPBB
 - Thossy, Michael, thosm1 (präsent)
 - michael.thossy@students.bfh.ch
 - Moser, Florian, mosef5 (präsent)
 - florian.moser@students.bfh.ch
 - Lupberger, Yanis, lupby1 (präsent)
 - yanisjoel.lupberger@students.bfh.ch
- rlm1-02 loT in the gastronomy sector at BFH Rolex Building
 - Fabio Bertagna, bertf3 (präsent)
 - fabio.bertagna@students.bfh.ch
 - Yannik Dällenbach, dally1 (präsent)
 - yannik.daellenbach@students.bfh.ch
 - Kunz, Philip, kunzp9 (präsent)
 - philipjonah.kunz@students.bfh.ch
- rlm1-05 Indoor Localization with Bluetooth Technologies
 - Zingg, Nicole, zingn1 (präsent)
 - nicole.zingg@students.bfh.ch
 - Pfeiffer, Thierry, pfeit2 (präsent)
 - thierry.pfeiffer@students.bfh.ch
 - Matthias, Santschi, santm6
 - matthiaslukas.santschi@students.bfh.ch
- Product Owner
 - @ Michael Röthlin (präsent)
 - michael.roethlin@bfh.ch

Diskussionsthemen

Zeit	Eintrag	Notizen
18:30	Eintreffen	• -

18:35	Begrüssung	 Frage an alle, ob Anforderungen Moser und andere bestätigt Verständnis Kunz nennt Diskussion punkto Stand "Format" vom letzten Meeting. Eigel erläutert: rlm1-04 stellt Service zur Verfügung und Publisher befüllen über REST-API. Röthlin begründet durch verschiedene Bedürfnisse der Einzelnen Gruppen. Weniger Objektmodellierung bei rlm1-04, mehr Funktionalität, tiefer. Corporate Design über mehrere Inkremente.
18:45	Frage an Röthlin	 Wäre ein "Muss" regelmässig Aktualisierungen zu veröffentlichen? Röthlin: Nein, Augenmerk auf saubere Arbeitsweise und Möglichkeit von HTTP-PUTs. Ganze Dashboards sind möglich, kein Zwang von Regelmässigkeiten von aktuellen Daten.
18:50	Resumé Anforderungswechsel	 Eigel: Scrum perfekt bzgl. Anforderungswechsel. Begründung lange Wartezeiten: Konzeptionierung dauerte über Schätzung. Bereitschaft regelmässigerer Meetings, grundsätzlich alle 2, 3 Wochen, bei Bedarf früher.
18:54	 Frage von Röthlin punkto Machbarkeit & Weitere Vorgehensweise 	 Forderungen von Anforderungen umsetzbar? Waser: Swagger UI ist schon veröffentlicht. Wird umfangreich umgesetzt. Röthlin: Wie möchten Publisher benachrichtigt? Kunz, Zingg, Moser: Wäre denkbar, dass wir bei Änderungen ein Mail versendet werden mit vielleicht Release Notes. Meetings sollen ausschliesslich auf Anfragen organisiert werden wegen Zeitverlust. Eigel: Einverständnis, Veröffentlichungen über Daumen gepeilt alle Sprints. Röthlin: Einverständnis. Kunz: Thema ist abstrakt.
18:59	 Frage Akzeptanz Anforderungen 	 Röthlin: Einsicht/Ausblick Backlog rlm1-04 und Möglichkeit von agiler Vorgehensweise. Input von Partnerteams in welchen Perioden? Eigel: Umgesetzt, Änderungen werden fortlaufend eingebettet und priorisiert.
19:03	Zusammenfassung	Röthlin: Projektziel wird angemessen umgesetzt, auf gutem Weg.
19:04	Verabschiedung	• -

✓ Handlungspunkte

I Entscheidungen

- Bei Änderungen von rlm1-04: Ein Mail versenden mit "Release Notes".
 Meetings ausschliesslich auf Anfragen organisieren wegen Zeitverlust.

13.2 Personas und Storyboards

13.2.1 Elma, BFH-Gebäude-Nutzer/BFH-Gebäude-Besucher (Low Level)

Persona: Elma, 55 Jahre alt

Beschreibung:

Elma ist 55 Jahre alt, hat 3 erwachsene Kinder und lebt mit Ihrem Mann in Wangen an der Aare. Sie arbeitet seit diesem Semester im Pensum von 60% am Empfang der BFH im SIPBB-Gebäude am Bieler Bahnhof. Durch etliche Gespräche am Mittagstisch in der Mensa des alten Rolex-Gebäudes, den stürmischen Anfragen der Studenten und einigen Weiterbildungen freundet Sie sich allmählich mit dem Techie-Fachjargon an. Ihr gefällt die Handhabung der Kompetenzen im Betrieb, denn Sie wird öfters gebeten in irgendwelchen von der BFH bereitgestellten Portalen kleinere Aufgaben zu erledigen. Sie gewöhnt sich schnell an das einheitliche Design der BFH.

Rolle:

- Mutter dreier erwachsener Kinder
- Teilzeit-Angestellte am Empfang der BFH (SIPBB-Gebäude)

Skills:

- Schnelle Auffassungsgabe

Ziele:

- Weiterhin sämtliche Ihr zugetragenen Aufgaben mit gewohntem Erfolg zu erledigen
- Mehr Eigeninitiative zeigen

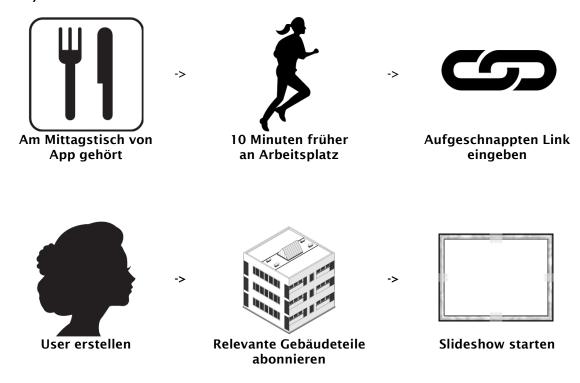
Pain points:

- Rissiger Geduldsfaden

Short Statements (Als [persona] möchte ich [feature], sodass [goal]):

 Als Elma möchte ich sehen, ob die neue InteractiveRoomMapDisplay-App, von der am Mittagstisch die Rede war was taugt, sodass ich meinen Mitarbeitern den Vorschlag bringen kann auch so ein Tablet an den Eingang zu stellen.

Storyboard zu Elma:



13.2.2 Guido, BFH-Gebäude-Besucher (High Level)

Persona: Guido, 23 Jahre alt

Beschreibung:

Guido ist 23 Jahre alt, kommt aus Karslruhe und macht ein Austausch-Semester an der BFH. Da er seinem Stil stets treu bleibt und alles im letzten Moment bewerkstelligt, so tat er dies auch bezogen auf die Organisation dieses Semesters. Er schnappte sich die erstbeste Wohnung und finanziert sich diese nun mit zwei Nebenjobs. Feiern darf natürlich nicht zu kurz kommen. Alles zusammen und das Vollgas.

Rolle:

- Austauschstudent BFH mit 2 Nebenjobs

Skills:

Solides Networking

Ziele:

- Effiziente Alltagsplanung

Pain points:

- Alles auf den letzten Drücker

Short Statements (Als [persona] möchte ich [feature], sodass [goal]):

- Als Guido möchte ich in meiner Zimmerstunde sehen, ob es sich lohnt vor dem Modulbesuch kurz das bald überfällige Modulbuchungsblatt im Sekretariat anzufordern, sodass ich nicht vergebens bis zu zwei Mal Schlange stehe.

Storyboard zu Guido:



Betreten BFH-Gebäude



Slideshow durchlaufen lassen bis Sekretariat auftaucht



Von Mitschüler gestört und Slide verpasst



Entdeckung Funktion Slide-Auswahl



Slide mit Sekretariat auswählen



Sekretariat ON FIRE -> also später versuchen

13.2.3 Juanito, Zukünftiges Entwicklerteam (HighLevel)

Persona: Juanito, 45 Jahre alt

Beschreibung:

Juanito ist 45 Jahre alt und lebt allein in seiner Goldküsten-Villa am Zürisee. Sollte Ihm langweilig werden, sieht er sich nach Investment-Möglichkeiten um. Besessen von der Idee, dass seine Angestellten allesamt einfach zu gestresst sind, sucht er gerne nach Möglichkeiten, um dem Abhilfe zu verschaffen, ohne den Ertrag seiner aufgekauften Firmen zu schmälern und Ihnen nicht zu sehr ins Tagesgeschäft zu funken.

Rolle:

- Privatier mit IT-Affinität

Skills:

Geschäftssinn

Ziele:

- Mehr Kapital

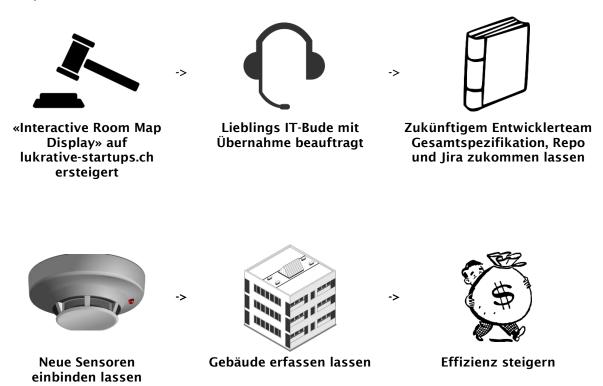
Pain points:

- Null-Toleranz bei Verzögerungen

Short Statements (Als [persona] möchte ich [feature], sodass [goal]):

- Als Juanito möchte ich nach der Übernahme des Projekts eine nachvollziehbare Architektur vorfinden, sodass eine allfällige Weiterentwicklung lukrativ bleibt.

Storyboard zu Juanito:



13.3 REST-API

13.3.1 Version 0.0.1 openapi: 3.0.3 info: title: Interactive Room Map Display description: The interactive room map display is used, to display various kinds of data about those room. version: 0.0.1 servers: - url: http://bfh-web-labs.ch/api tags: - name: room description: Access and operations about rooms - name: object description: Access and operations about objects - name: mensa description: Access and operations about mensas paths: /room/{building}/{floor}: put: tags: - room summary: Update room status description: Update room status by coordinate, floor and building operationId: updateRoom parameters: - name: building in: path description: name of building required: true schema: type: string format: string - name: floor in: path description: name of floor required: true schema: type: string format: string - name: coordinate in: query description: coordinate of room required: true schema: type: string format: string - name: status in: query

description: status of room

required: true schema:

responses:

type: boolean format: boolean

```
'200':
      description: successful operation
     '400':
      description: Invalid input
     '401':
      description: Invalid credentials supplied
     '500':
      description: Internal server error
/object/{building}/{name}:
 put:
  tags:
   - object
  summary: Update object status
  description: Update object status by coordinate, floor and building
  operationId: updateObject
  parameters:
   - name: building
     in: path
     description: name of building
     required: true
     schema:
      type: string
   - name: name
     in: path
     description: name of object
     required: true
     schema:
      type: string
   - name: floor
     in: query
     description: name of floor
     required: true
     schema:
      type: string
   - name: coordinate
     in: query
     description: coordinate of object
     required: true
     schema:
      type: string
  responses:
     '200':
      description: successful operation
     '400':
      description: Invalid input
     '401':
      description: Invalid credentials supplied
     '500':
      description: Internal server error
/mensa/{building}/{floor}:
 put:
  tags:
   - mensa
  summary: Update mensa status
  description: Update mensa status by coordinate, floor and building
  operationId: updateMensa
```

parameters: - name: building in: path description: name of building required: true schema: type: string - name: floor in: path description: name of floor required: true schema: type: string - name: coordinate in: query description: coordinate of mensa required: true schema: type: string - name: waitingTime in: query description: waiting time in mensa required: true schema: type: integer format: int64 - name: frequency in: query description: frequency in mensa required: true schema: type: integer format: int64 responses: '200': description: successful operation '400': description: Invalid input '401': description: Invalid credentials supplied '500': description: Internal server error

13.3.2 Version 1.0

```
openapi: 3.0.3
info:
 title: Interactive Room Map Display
 description: The interactive room map display is used, to display various kinds of data about several
rooms. This v1 uses a low-level approach.
 version: '1.0'
servers:
 - url: https://service.p1-irm.bfh-web-labs.ch/api/v1/
tags:
 - name: users
  description: Information and operations related to users
 - name: buildings
  description: Information and operations related to buildings
 - name: views
  description: Information and operations related to views
paths:
 /login:
  post:
   tags:
     - users
   summary: Login
   description: Login into the service
   operationId: login
   requestBody:
     description: id and hashed password of user in JSON format
     required: true
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/Login'
   responses:
     '200':
      description: returns an authentication token
      content:
       application/json:
         schema:
          $ref: '#/components/schemas/AuthToken'
     '401':
      description: bad credentials
      description: user not allowed to access resource
      description: resource not found
      description: method not allowed
     '500':
      description: internal server error
 /buildings:
  get:
   tags:
     - buildings
   summary: Returns all buildings
   description: Returns a list of all registered buildings with id and name
   operationId: getBuildings
   responses:
```

```
'200':
     description: returns a JSON array with all buildings
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/Buildings'
   '401':
     description: user unauthicated
   '403':
     description: user not allowed to access resource
     description: resource not found
   '405':
     description: method not allowed
   '500':
     description: internal server error
  security:
   - bearerAuth: []
/buildings/{id}/floors:
 get:
  tags:
   - buildings
  summary: Returns floors from building
  description: Returns all floors from a specific building
  operationId: getFloors
  parameters:
   - name: id
     in: path
     description: id of building
     required: true
     schema:
      type: integer
      format: int64
  responses:
    '200':
     description: returns a JSON array with all floors
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/Floors'
   '401':
     description: user unauthicated
     description: user not allowed to access resource
     description: resource not found
   '405':
     description: method not allowed
   '500':
     description: internal server error
  security:
   - bearerAuth: []
/views:
 put:
  tags:
   - views
```

```
summary: Send view
    description: Send view to service
    operationId: sendView
    requestBody:
     description: id, building, floor and HTML of view
     required: true
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/View'
    responses:
     '201':
      description: resource created
     '401':
      description: user unauthicated
      description: user not allowed to access resource
     '404':
      description: resource not found
     '405':
      description: method not allowed
     '500':
      description: internal server error
    security:
     - bearerAuth: []
components:
 schemas:
  Login:
    type: object
    properties:
     email:
      type: string
      format: string
      example: 'mensa@bfh.ch'
     password:
      type: string
      format: string
      example: 'YourPasswordGoesHere'
  AuthToken:
    type: object
    properties:
     email:
      type: string
      format: string
      example: 'mensa@bfh.ch'
     token:
      type: string
      format: string
      example: 'JKij56lelN4aldcenesaAKXBWKaercvc'
  View:
   type: object
    properties:
     name:
      type: string
      format: string
      example: 'mensa-rolex'
```

```
building_id:
     type: integer
     format: int64
     example: 2
   floor_id:
     type: integer
     format: int64
     example: 3
   html:
     type: string
     format: string
     example: '<html>...</html>'
 Buildings:
  type: array
  items:
   type: object
   properties:
     id:
      type: integer
      format: int64
      example: 1
     name:
      type: string
      format: string
      example: 'SIPBB'
 Floors:
  type: array
  items:
   type: object
   properties:
      type: integer
      format: int64
      example: 1
     name:
      type: string
      format: string
      example: 'SIP-S1'
securitySchemes:
 bearerAuth:
  type: http
  scheme: bearer
```

13.3.3 Version 1.1

```
openapi: 3.0.3
info:
 title: Interactive Room Map Display
 description: The interactive room map display is used, to display various kinds of data about several
rooms. This v1 uses a low-level approach.
 version: '1.1'
servers:
 - url: https://rlm1-04.p1-irm.bfh-web-labs.ch/api/
tags:
 - name: users
  description: Information and operations related to users
 - name: buildings
  description: Information and operations related to buildings
 - name: views
  description: Information and operations related to views
paths:
 /login:
  post:
   tags:
     - users
    summary: Login
    description: Login into the service
    operationId: login
    requestBody:
     description: email and password of user in JSON format
     required: true
     content:
      application/json:
        schema:
         $ref: '#/components/schemas/Login'
    responses:
     '201':
      description: returns an authentication token
      content:
        application/json:
         schema:
          $ref: '#/components/schemas/AuthToken'
     '400':
      description: invalid accept header
     '401':
      description: wrong email or password
     '404':
      description: resource not found
     '405':
      description: method not allowed
     '415':
      description: unsupported media type
     '429':
      description: too many requests
 /buildings:
  get:
    tags:
     - buildings
    summary: Returns all buildings
    description: Returns a list of all registered buildings with id and name
```

```
operationId: getBuildings
  responses:
   '200':
     description: returns a JSON array with all buildings
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/Buildings'
   '400':
     description: invalid accept header
   '401':
     description: user unauthicated
   '404':
     description: resource not found
   '405':
     description: method not allowed
   '429':
     description: too many requests
  security:
   - bearerAuth: []
/buildings/{id}/floors:
 get:
  tags:
   - buildings
  summary: Returns floors from building
  description: Returns all floors from a specific building
  operationId: getFloors
  parameters:
   - name: id
     in: path
     description: id of building
     required: true
     schema:
      type: integer
      format: int64
  responses:
   '200':
     description: returns a JSON array with all floors
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/Floors'
   '400':
     description: invalid accept header
     description: user unauthicated
   '404':
     description: resource not found
   '405':
     description: method not allowed
     description: too many requests
  security:
    - bearerAuth: []
/views:
 put:
```

```
tags:
     - views
   summary: Send view
   description: Send view to service
   operationId: sendView
   requestBody:
     description: name, building, floor and HTML of view
     required: true
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/View'
   responses:
     '201':
      description: resource created
      content:
       application/json:
         schema:
          $ref: '#/components/schemas/ViewResponse'
     '400':
      description: invalid accept header
     '401':
      description: user unauthicated
     '403':
      description: user not authorized to update this view
     '404':
      description: resource not found
      description: method not allowed
      description: unsupported media type
     '429':
      description: too many requests
   security:
     - bearerAuth: []
components:
 schemas:
  Login:
   type: object
   properties:
     email:
      type: string
      format: string
      example: 'mensa@bfh.ch'
     password:
      type: string
      format: string
      example: 'YourPasswordGoesHere'
  AuthToken:
   type: object
   properties:
     email:
      type: string
      format: string
      example: 'mensa@bfh.ch'
     token:
```

 $rlm 1-04-Interactive Room Map Display-Gesamt spezifikation\ v3-0-Eigel Fuhrer Waser. docx,\ Version\ 3.0,\ 20.01.2023$

```
type: string
   format: string
   example: '1|61UQPSKeb8VXUr2xMnXiQtvFLfjdUCYdiMt1A6Yd'
View:
 type: object
 properties:
  name:
   type: string
   format: string
    example: 'rolex-mensa'
  building_id:
   type: integer
   format: int64
    example: 1
  floor_id:
   type: integer
   format: int64
   example: 1
  view:
   type: string
   format: string
   example: '<html>...</html>'
Buildings:
 type: array
 items:
  type: object
  properties:
   id:
     type: integer
     format: int64
     example: 1
    name:
     type: string
     format: string
     example: 'SIPBB'
Floors:
 type: array
 items:
  type: object
  properties:
   id:
     type: integer
     format: int64
     example: 1
    name:
     type: string
     format: string
     example: 'SIP-S1'
ViewResponse:
 type: object
 properties:
  name:
   type: string
   format: string
   example: 'rolex-mensa'
  building_id:
```

```
type: integer
      format: int64
      example: 1
     floor_id:
      type: integer
      format: int64
      example: 1
     view:
      type: string
      format: string
      example: '<html>...</html>'
     updated_at:
      type: string
      format: string
      example: '2022-12-21T11:49:52.000000Z'
 securitySchemes:
  bearerAuth:
   type: http
    scheme: bearer
                           13.3.4 Version 1.2
openapi: 3.0.3
info:
 title: Interactive Room Map Display
 description: The interactive room map display is used, to display various kinds of data about several
rooms. This v1 uses a low-level approach.
 version: '1.2'
servers:
 - url: https://service.p1-irm.bfh-web-labs.ch/api/v1/
tags:
 - name: users
  description: Information and operations related to users
 - name: buildings
  description: Information and operations related to buildings
 - name: views
  description: Information and operations related to views
paths:
 /login:
  post:
    tags:
     - users
   summary: Login
    description: Login into the service
    operationId: login
    requestBody:
     description: email and password of user in JSON format
     required: true
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/Login'
    responses:
     '201':
      description: returns an authentication token
      content:
```

```
application/json:
       schema:
         $ref: '#/components/schemas/AuthToken'
   '400':
     description: invalid accept header
   '401':
     description: wrong email or password
     description: resource not found
   '405':
     description: method not allowed
   '415':
     description: unsupported media type
   '422':
     description: bad formed request body
     description: too many requests
/buildings:
 get:
  tags:
   - buildings
  summary: Returns all buildings
  description: Returns a list of all registered buildings with id and name
  operationId: getBuildings
  responses:
   '200':
     description: returns a JSON array with all buildings
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/Buildings'
   '400':
     description: invalid accept header
   '401':
     description: user unauthicated
   '404':
     description: resource not found
   '405':
     description: method not allowed
   '429':
     description: too many requests
  security:
   - bearerAuth: []
/buildings/{id}/floors:
 get:
  tags:
   - buildings
  summary: Returns floors from building
  description: Returns all floors from a specific building
  operationId: getFloors
  parameters:
   - name: id
     in: path
     description: id of building
     required: true
     schema:
```

```
type: integer
      format: int64
  responses:
   '200':
     description: returns a JSON array with all floors
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/Floors'
   '400':
     description: invalid accept header
   '401':
     description: user unauthicated
   '404':
     description: resource not found
     description: method not allowed
   '429':
     description: too many requests
  security:
   - bearerAuth: []
/views:
 put:
  tags:
   - views
  summary: Send view
  description: Send view to service
  operationId: sendView
  requestBody:
   description: name, building, floor and HTML of view
   required: true
   content:
     application/json:
      schema:
       $ref: '#/components/schemas/View'
  responses:
   '201':
     description: resource created
     content:
      application/json:
       schema:
         $ref: '#/components/schemas/ViewResponse'
   '400':
     description: invalid accept header
   '401':
     description: user unauthicated
   '403':
     description: user not authorized to update this view
   '404':
     description: resource not found
   '405':
     description: method not allowed
     description: unsupported media type
   '422':
     description: bad formed request body
```

 $rlm 1-04-Interactive Room Map Display-Gesamt spezifikation\ v3-0-Eigel Fuhrer Waser. docx,\ Version\ 3.0,\ 20.01.2023$

```
'429':
      description: too many requests
   security:
    - bearerAuth: []
components:
 schemas:
  Login:
   type: object
   properties:
     email:
      type: string
      format: string
      example: 'mensa@bfh.ch'
     password:
      type: string
      format: string
      example: 'YourPasswordGoesHere'
  AuthToken:
   type: object
   properties:
    email:
      type: string
      format: string
      example: 'mensa@bfh.ch'
     token:
      type: string
      format: string
      example: '1|61UQPSKeb8VXUr2xMnXiQtvFLfjdUCYdiMt1A6Yd'
  View:
   type: object
   properties:
     name:
      type: string
      format: string
      example: 'rolex-mensa'
     building_id:
      type: integer
      format: int64
      example: 1
     floor_id:
      type: integer
      format: int64
      example: 1
     view:
      type: string
      format: string
      example: 'This is some <strong>HTML</strong> content.<h3>hello</h3><img
src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAUAAAAFCAYAAACNbyblAAAAHEIEQV
QI12P4//8/w38GIAXDIBKE0DHxgljNBAAO9TXL0Y4OHwAAAABJRU5ErkJggg==" alt="Italian Trulli"><img
src="https://example.com/image.jpg" alt="Example Image"><script>alert("Hello, world!");</script>'
  Buildings:
   type: array
   items:
     type: object
     properties:
      id:
```

```
type: integer
       format: int64
       example: 1
      name:
       type: string
       format: string
       example: 'SIPBB'
  Floors:
   type: array
   items:
    type: object
    properties:
      id:
       type: integer
       format: int64
       example: 1
      name:
       type: string
       format: string
       example: 'SIP-S1'
  ViewResponse:
   type: object
   properties:
    name:
      type: string
      format: string
      example: 'rolex-mensa'
    building_id:
      type: integer
      format: int64
      example: 1
    floor_id:
      type: integer
      format: int64
      example: 1
    view:
      type: string
      format: string
      example: 'This is some <strong>HTML</strong> content.<h3>hello</h3><img
src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAUAAAAFCAYAAACNbyblAAAAHEIEQV
QI12P4//8/w38GIAXDIBKE0DHxgljNBAAO9TXL0Y4OHwAAAABJRU5ErkJggg==" alt="Italian Trulli">'
    updated_at:
      type: string
      format: string
      example: '2022-12-21T11:49:52.000000Z'
 securitySchemes:
  bearerAuth:
   type: http
   scheme: bearer
```