**Anforderungsmanagement | 01**

**Definition und Aufgaben des Funktionsmasters (Buchungssystem|Konferenzräume):**

1. **Definition des Funktionsmasters**

Der Funktionsmaster ist ein zentraler Bestandteil der Softwarearchitektur, der die Logik und die Bedingungen definiert, unter denen bestimmte Funktionen oder Module innerhalb eines Systems ausgeführt werden. Dies kann insbesondere in komplexen Anwendungen relevant sein, in denen verschiedene Geschäftslogiken berücksichtigt werden müssen.

1. **Aufgaben des Funktionsmasters**

**Regeldefinition:** Der Funktionsmaster legt fest, welche Bedingungen erfüllt sein müssen, damit bestimmte Funktionen aktiviert oder deaktiviert werden. Dies kann Logik, Zustände, Parameter oder andere relevante Faktoren umfassen.

**Koordination:** Er koordiniert die Interaktion zwischen verschiedenen Komponenten und Modulen des Systems und stellt sicher, dass die Ausführung der Funktionen reibungslos erfolgt.

**Dokumentation:** Der Funktionsmaster dokumentiert alle relevanten Bedingungen und Logiken, sodass die Entwickler und Tester nachvollziehen können, wie und warum bestimmte Funktionen implementiert wurden.

**Rechtliche Verbindlichkeiten:**

**„Muss“ (Must):**

**Definition:** Der Begriff „muss“ wird verwendet, um Anforderungen zu kennzeichnen, die zwingend erfüllt werden müssen.

Dies bedeutet, dass die Nichteinhaltung dieser Anforderung zu einem vollständigen Mangel an Funktionalität

oder zu rechtlichen Konsequenzen führen kann.

**Beispiele:**

* Der Raum muss verfügbar sein, damit eine Buchung erstellt werden kann
* Der Raum muss einen Namen bzw. eine bestimmte Kennzeichnung haben, dass er einmalig und eindeutig identifizierbar ist
* Die Buchung muss im System hinterlegt sein und einen bestimmten Zeitraum belegen
* Bei Stornierung muss der Raum wieder als verfügbar angezeigt werden

**"Sollte" (Should):**

**Definition:** „Sollte“ wird verwendet, um Anforderungen zu beschreiben, die wünschenswert, aber nicht zwingend sind.

Die Nichterfüllung dieser Anforderungen könnte zwar zu einem weniger optimalen Produkt führen,

jedoch sind sie nicht kritisch für die grundlegende Funktionalität.

**Beispiele:**

* Die Software sollte beinhalten, dass ein Raum nur stornierbar, wenn es mehr als 24h zuvor ist
* Jeder Raum sollte mit Details wie Kapazität, Ausstattung und Lage angezeigt werden/ausgestattet sein
* Die Software sollte beinhalten, dass bei einem reservierten Raum die Teilnehmerzahl, der Zweck und der Name der Person die reserviert hat, angezeigt wird

**3.3 "Wird" (Will)**

**Definition:** „Wird“ wird verwendet, um zukünftige Funktionalitäten oder Entwicklungen zu beschreiben,

die in einer bestimmten Phase des Projekts implementiert werden sollen. Dies kann auch als Absichtserklärung betrachtet werden.

**Beispiel:**

* Die Anwendung wird eine Benachrichtigung an eingeladene Teilnehmer der Raumbuchung herausschicken
* Die Anwendung wird eine graphische Ansicht über die Räume verfügen
* Die Anwendung wird auf Basis der Teilnehmerzahl und Art des Meetings Räume vorschlagen können

**Funktionalen Anforderungen auf Basis der Aufgabenstellung:**

* **Raumverfügbarkeit prüfen**: Das System muss es Nutzern ermöglichen, die Verfügbarkeit von Konferenzräumen zu einem bestimmten Zeitpunkt zu überprüfen.
* **Buchungen erstellen/verwalten**: Nutzer müssen Konferenzräume für bestimmte Zeitfenster buchen können.
* **Stornierungen**: Nutzer sollen Buchungen stornieren können, wenn sie den Raum nicht mehr benötigen.

**den präzise formulierten Anforderungen (Schritte 0-5) auf Basis der Aufgabenstellung:**

1. **Raumverfügbarkeit prüfen**:

**Schritt 0:** Das System: BuchungsSys

BuchungsSys.

**Schritt 1:** Die rechtliche Verbindlichkeit: MUSS

BuchungsSys MUSS.

**Schritt 2:** Das Prozesswort: prüfen

BuchungsSys MUSS prüfen.

**Schritt 3:** Die Art der Funktionalität: Interaktion mit Datenbank

BuchungsSys MUSS den Inhalt einer Datenbank prüfen können.

**Schritt 4:** Das Objekt: Konferenzräume

BuchungsSys MUSS die Konferenzräume die in einer Datenbank enthalten sind prüfen können.

**Schritt 5:** Die Bedingungen: Zeitraum: Buchungssystem angeschaltet (läuft)

SOLANGE BuchungsSys läuft, MUSS BuchungsSys die Konferenzräume die in einer Datenbankenthalten sind prüfen können.

1. **Buchungen erstellen/verwalten**:

**Schritt 0:** Das System: BuchungsSys

BuchungsSys.

**Schritt 1:** Die rechtliche Verbindlichkeit: MUSS

BuchungsSys MUSS.

**Schritt 2:** Das Prozesswort: erstellen

BuchungsSys MUSS erstellen.

**Schritt 3:** Die Art der Funktionalität: Benutzerinteraktion mit Rolle Benutzer

BuchungsSys MUSS dem Benutzer die Möglichkeit bieten eine Buchung zu erstellen.

**Schritt 4:** Das Objekt: Konferenzraum

BuchungsSys MUSS dem Benutzer die Möglichkeit bieten eine Buchung für einen Konferenzräume zu erstellen.

**Schritt 5:** Die Bedingungen: Zeitraum: Ausführen der Funktion Konferenzraum buchen

Solange BuchungsSys die Funktion Konferenzraum buchen ausführt, MUSS BuchungsSys dem Benutzer die Möglichkeit bieten eine Buchung für einen Konferenzraum zu erstellen.

1. **Stornierungen**:

**Schritt 0:** Das System: BuchungsSys

BuchungsSys.

**Schritt 1:** Die rechtliche Verbindlichkeit: MUSS

BuchungsSys MUSS.

**Schritt 2:** Das Prozesswort: stornieren

BuchungsSys MUSS stornieren.

**Schritt 3:** Die Art der Funktionalität: Interaktion mit Datenbank

BuchungsSys MUSS den Inhalt einer Datenbank stornieren können.

**Schritt 4:** Das Objekt: Konferenzräume

BuchungsSys MUSS einen Konferenzraum einer Datenbank stornieren können.

**Schritt 5:** Die Bedingungen: Zeitpunkt: mehr als 24h vor dem Zeitpunkt der Nutzung

SOLANGE der Zeitpunkt der Nutzung noch mehr als 24h in der Zukunft liegt, MUSS BuchungsSys einen Konferenzraum der Datenbank stornieren können.