

6.12.4. Create Tenant

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Tenant-Registrierungsseite	Anzeige und Weiterleitung der Eingaben
Tenant-Registrierung	Nimmt die Eingaben an und registriert den Tenant in der Datenbank (über die Komponente Datenbank-Zugriff)
Tenant-Liste (in der Datenbank)	Verwaltet die Tenants
Datenbank-Zugriff	Bietet Zugriff auf die Datenbank
Tenant-Registrierungsdatenüberprüfung	Überprüft die Daten
Benutzerverwaltung	Setzt den User als Tenant-Admin

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Verfügbarkeit des Tenantnamens (Boolean)	Tenant-Registrierungsdatenüberprüfung (über die Tenantliste in der Datenbank)
Eingabefelder (siehe Tenant-Registrierung)	Tenant-Registrierungsseite

1.3. Welche Eingabe-/Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Tenant-Name (String)	Tenant-Liste (Datenbank)	Workitems-Seite Workflow-Models-Seite Workflow-Instance-Seite Tenant-Switch-Seite Tenant-Übersicht (Superadmin) Tenant-Einstellungen

Tenant-Beschreibung (String)	s.o.	s.o.
------------------------------	------	------

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Verfügbarkeit Tenant-Name (Boolean)	Wird on-the-fly ermittelt	

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
Tenant-Registrierungsseite, Tenant-Registrierung	Weiterleitung Tenant-Eingabedaten
Tenant-Registrierung, Datenbank-Zugriff	Funktionsaufruf (registerTenant) mit Tenant-Daten als Parameter
Tenant-Registrierungsseite, Tenant-Registrierungsüberprüfung	Weiterleitung des Tenant-Names
Tenant-Registrierungsüberprüfung, Datenbank-Zugriff	Funktionsaufruf (isAvailable) mit Tenant-Namen als Parameter
Tenant-Registrierung, Benutzerverwaltung	Funktionsaufruf (Nachricht): Benutzer ist Tenant-Admin vom gerade erstellten Tenant

6.12.5 Switch Tenant

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Tenantmitgliedschafts-Liste des Benutzers (aus der Benutzerverwaltung)	Verwaltet die Tenant-Mitgliedschaften des Benutzers
Benutzereinstellungen (welcher Tenant ist als „aktuell“ gesetzt?)	Verwaltet die Benutzereinstellungen
Tenant-Logo und –Farbschemaliste (aus der Datenbank)	Speichert und verwaltet die Tenant-Logos und Farbeinstellungen
Logo- und Farbschemawechsler	Verändert das Aussehen und Logo der Website bei einem Tenant-Switch
Tenant-Switchseite	Zeigt den aktuellen Tenant an und leitet den ausgewählten Tenant weiter (an Logo- und Farbschemawechsler)

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Tenant-Liste des Benutzers	Benutzerverwaltung

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Zieltenant (String)	Benutzereinstellungen	Workitem-List Logo- und Farbschemawechsler

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
Tenantswitch-Seite, Logo- und Farbschemawechsler	Weiterleitung des Tenantnamens
Tenantswitch-Seite Benutzereinstellungen ,	Weiterleitung des Tenantnamens
Tenant-Logo und –Farbschemaliste, Logo- und Farbschemawechsler	Hole Daten aus der Datenbank Logo und Farbschema aus Tenant-Logo und –Farbschemaliste.

Change forgotten account password

VORSICHT: DAU kann zweimal "forgot pwd" klicken ohne 1. PWD zu bestätigen

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Benutzerverwaltung	Generiert Zufallspasswort, Überprüft Verfügbarkeit des Benutzers
Browser/UI	Zeigt Fehlermeldung, Erfolgsmeldung, Eingabeformular
Koordinator	Koordiniert andere Komponenten
E-Mail-WS	Verschickt E-Mail
Authentifizierer	Generiert Passwort-Hash

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Passwort	Benutzerverwaltung
Benutzername	Benutzerverwaltung

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
E-Mail Adresse (String)	Keiner (User)	E-Mail-WS, Benutzerverwaltung

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
-------------	-------------	-------------

Passwort (String)	Keiner	Authentifizierer, E-Mail-WS
Benutzername (String)	Datenbank	E-Mail-WS

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	E-Mail-Adresse Return: Erfolgsstatus
Koordinator -> Benutzerverwaltung	createNewPassword(E-Mail-Adresse) Return: Passwort+Benutzername/Exception
Benutzerverwaltung -> Authentifizierer	generatePWHash(Passwort) Return: Hash (String)
Benutzerverwaltung -> Datenbank	Passwort-Hash speichern
Koordinator -> E-Mail-WS	E-Mail-Daten, Passwort, Benutzername Return: Erfolgsstatus

Confirm tenant membership invitation

!!!Siehe Anmerkungen zu den Use-Cases 6.2.7 und 6.4.4!!!

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Benutzerverwaltung	Prüft ob Registrierungs-Link noch gültig ist & nicht disabled; Benutzerdaten verfügbar machen
Browser/UI	Zeigt Seite mit „confirm button“ & Statusmeldung
User-Tenant-Mapper	Fügt Benutzer zum Tenant hinzu (und umgekehrt)
Koordinator	Koordiniert Komponenten
E-Mail-WS	Sends confirmation message
Workflow Management	Starte Workflow

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Registration link	E-Mail
Status	Koordinator
Tenant-Name	Benutzerverwaltung

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo sind sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Bestätigung (boolean)	keiner	Tenant Management
Registration link	keiner	Benutzerverwaltung, Browser/UI

(String)		
----------	--	--

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Tenant-Name (String)	DB	Browser/UI
Status-E-Mail (E-Mail)	keiner	E-Mail-WS

1.4.Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	Invite-ID aus dem Link Return: show confirmation/error page/goto use case „Register account as normal user“
Koordinator -> Benutzerverwaltung	checkInvite(Invite-ID) Return: user bzw. null/Exception (enthält Details: Link expired/user disabled)
UI -> Koordinator	confirm/reject Return: (un-)successful
Koordinator -> Benutzerverwaltung	lock(Invite-ID) (<- Schutz vor Doppel-Entscheidung) Return: success/error
Koordinator -> User-Tenant-Mapper	createMapping(UserID, TenantID)
Koordinator -> Benutzerverwaltung	deleteInvite(Invite-ID)
Koordinator -> Workflow Management	userRegistered(UserID) (<- asynchron: workflows starten, bei denen alle Benutzer jetzt registriert sind)
Koordinator -> E-Mail-WS	Sende E-Mail and admin mit confirmation status

Confirm workflow administration invitation

!!!Siehe Anmerkungen zu den Use-Cases 6.2.7 und 6.4.4!!!

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Benutzerverwaltung	Prüft ob Registrierungs-Link noch gültig ist & nicht disabled; Benutzerdaten verfügbar machen
Browser/UI	Zeigt Seite mit „confirm button“ & Statusmeldung
User-Workflow-Mapper	Fügt Benutzer zum Tenant hinzu (und umgekehrt)
Koordinator	Koordiniert Komponenten
E-Mail-WS	Sends confirmation message
Workflow Management	Starte Workflow; setze user als Instanz-Admin

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Registration link	E-Mail
Status	Koordinator
Tenant-Name	Benutzerverwaltung

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo sind sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Bestätigung (boolean)	keiner	Tenant Management

Registration link (String)	keiner	Benutzerverwaltung, Browser/UI
----------------------------	--------	--------------------------------

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Tenant-Name (String)	DB	Browser/UI
Status-E-Mail (E-Mail)	keiner	E-Mail-WS

1.4.Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	Invite-ID aus dem Link Return: show confirmation/error page/goto use case „Register account as normal user“
Koordinator -> Benutzerverwaltung	checkInvite(Invite-ID) Return: user bzw. null/Exception (enthält Details: Link expired/user disabled)
UI -> Koordinator	confirm/reject Return: (un-)successful
Koordinator -> Benutzerverwaltung	lock(Invite-ID) (<- Schutz vor Doppel-Entscheidung) Return: success/error
Koordinator -> User-Tenant-Mapper	createMapping(UserID, TenantID) (<- könnte schon existieren)
Koordinator -> Workflow Management	setInstanceAdmin(UserID, InstanceID)
Koordinator -> Benutzerverwaltung	deleteInvite(Invite-ID)
Koordinator -> Workflow Management	userRegistered(UserID) (<- asynchron: workflows starten, bei denen alle Benutzer jetzt registriert sind)
Koordinator -> E-Mail-WS	Sende E-Mail and admin mit confirmation status

Change e-mail address

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Profil anzeigen
User-Management-Komponente	Daten des Benutzers verfügbar machen/speichern/validieren
E-Mail-WS	Confirmation E-Mail senden
Koordinator	Koordiniert

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Profil-Daten	User-Management
Confirmation Link	E-Mail

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Neue E-Mail Adresse (String)	DB (user Eintrag, „unconfirmed Email“ Feld)	E-Mail-WS, User-Management-Komponente
Confirmation ID (String)	DB (user Eintrag, „confirmation ID“ Feld)	User-Management-Komponente

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Confirmation Link (HTTP-Link)	keiner	E-Mail-WS, Browser/UI

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	getUser(SessionID) Return: Benutzerprofildaten
Koordinator -> User Management	getUser(SessionID) Return: User
UI -> Koordinator	changeEmail(newEmail, SessionID) Return: success/failure
Koordinator -> User Management	changeEmail(newEmail, SessionID) Return: success/failure
User Management -> Koordinator	sendConfirmationMail(confirmation link, userName) Return: success/failure
Koordinator -> E-Mail-WS	„Send Email“ mit confirmation link und spreche Benutzer direkt an Return: success/failure
UI -> Koordinator	confirmEmail(confirmationID) Return: show success/failure
Koordinator -> User Management	confirmEmail(confirmationID) Return: success/failure (link validity)

Change password

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Koordinator	Koordiniert
User Management	Stellt Profil-Daten bereit & ändert Passwort
Browser/UI	Zeigt Daten an und nimmt Eingaben entgegen
Authentifizierer	Generiert Passwort-Hash

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Profil-Daten	User-Management

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Altes Passwort	keiner	Authentifizierer
Neues Passwort	keiner	Authentifizierer
Confirmation Passwort	keiner	UI (serverseitig)

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	changePassword(SessionID, newPassword, oldPassword) Return: success/Exception
Koordinator -> Authentifizierer	generateHash(newPassword)

	Return: newHash
Koordinator -> Authentifizierer	generateHash(oldPassword) Return: oldHash
Koordinator -> User Management	changePassword(SessionID, oldHash, newHash) Return: success/Exception

Change address data

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Koordinator	Koordiniert
User Management	Stellt Profil-Daten bereit & ändert Passwort
Browser/UI	Zeigt Daten an und nimmt Eingaben entgegen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Profil-Daten	User-Management

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
First Name (String)	DB	User Management
Last Name (String)	DB	User Management
Postal Code (String)	DB	User Management
Street (String)	DB	User Management
City (String)	DB	User Management

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	changeAddress(forename, surname, poCode, street, city, SessionID)

	Return: success/Exception
Koordinator -> User Management	changeAddress(forename, surname, poCode, street, city, SessionID) Return: success/Exception

Cancel Membership

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Benutzerverwaltung	Benutzerdaten verfügbar machen, prüft ob Benutzer Tenant-/Workflow-Admin ist, Benutzeraccount löschen
Browser/UI	Zeigt Seite mit „confirm button“ & Statusmeldung
Koordinator	Koordiniert Komponenten
Email-WS	Sendet Email an Tenant Admin

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Benutzerprofilaten	Benutzerverwaltung

~~1.3. Welche Eingabe / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?~~

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	deleteUser(SessionID) return: Bestätigung/Fehlermeldung
Koordinator -> Benutzerverwaltung	deleteUser(SessionID) return: Bestätigung/Fehlermeldung
Koordinator -> Email-WS	Notification an Tenant-Admin Senden

Log out of account

Wie machen wir das Session-Management?

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Zeigt „log out“ link und Welcome Page

Set User Status to Available

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Benutzerverwaltung	Benutzerdaten verfügbar machen, Benutzerstatus ändern
Browser/UI	Zeigt Profildaten an
Koordinator	Koordiniert Komponenten

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Benutzerprofildaten	Benutzerverwaltung

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
User Available (bool)	Keiner	UI, Koordinator, Benutzerverwaltung

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	setUserAvailableStatus(availableStatus)
Koordinator -> Benutzerverwaltung	setUserAvailableStatus(availableStatus)

Set User Status to Unavailable

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Benutzerverwaltung	Benutzerdaten verfügbar machen, Benutzerstatus ändern
Browser/UI	Zeigt Profildaten an
Koordinator	Koordiniert Komponenten

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Benutzerprofildaten	Benutzerverwaltung

1.3. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	setUserAvailableStatus(availableStatus)
Koordinator -> Benutzerverwaltung	setUserAvailableStatus(availableStatus)

6.3.4 Resume working on work item

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workitem-List-Seite	Zeigt Workitem-List an
Work-Item-List	Liste der Workitems mit den Items und Eigenschaften. WMS greift auf diese zu.
Workitem-Management-System (WIMS)	Verwaltet die Work-Items. <ul style="list-style-type: none"> - erstelle workitem - hole workitem - setze workitem status - lösche Workitem
Work-Item-Seite	Bearbeiten von Workitems (Datei hochladen, Eingabefelder)
Storage Service	Permanentes speichern von Dateien
Eingabvalidierung	Überprüft die Gültigkeit der Eingaben

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Work-Item-List	Workitem-Management-System
Gespeicherte Daten des Work-Items	Work-Item-List über das Workitem-Management-System

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Dynamisch, je nach Workitem	Work-Item Objekt in der DB	WMS (Human-Task-Webservice)

		Work-Item-Seite
--	--	-----------------

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Work-Item Objekt	DB	WMS

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
Workitem-List-Seite, Workitem-Management-System	Abfrage der Workitem-List des Benutzers beim WMS
Workitem-Management-System , Work-Item-List	WMS holt die Workitemlist aus der DB
Workitem-Management-System (WMS), Storage Service	WMS ruft storage service mit folgenden Übergabeparametern auf: Dateiname, Rechte
Work-Item-Seite, Eingabvalidierung	Seite schickt Daten zur Validierung an Eingabvalidierung.

Change User Roles

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workflow Instance Configurator	Rollen der Workflowbenutzer ändern
Browser/UI	Zeigt die Workflowteilnehmer samt Rollen an
Koordinator	Koordiniert Komponenten

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflowbenutzer und -rollen	Workflow Instance Configurator

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Liste mit Usern und zugehörigen Rollen (UserId/Enum)	Workflow Instance Configurator (DB)	UI, Koordinator, Workflow Instance Configurator

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Liste mit Usern und zugehörigen Rollen (UserId/Enum)	Workflow Instance Configurator (DB)	UI, Koordinator, Workflow Instance Configurator

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	Liste mit Benutzern und Rollen speichern
Koordinator -> Workflow Instance Configurator	Liste mit Benutzern und Rollen

Add User to a Workflow Instance

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workflow Instance Configurator	Workflowteilnehmer mit Rolle hinzufügen
Benutzerverwaltung	Benutzername/Emailadresse verifizieren
Browser/UI	Zeigt die Workflowteilnehmer samt Rollen an
Koordinator	Koordiniert Komponenten

1.2. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Username/Emailadresse (String)	Workflow Instance Configurator (DB)	UI, Koordinator, Workflow Instance Configurator
Rolle (Enum?)	Workflow Instance Configurator (DB)	UI, Koordinator, Workflow Instance Configurator

1.3. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	addWorkflowUser(WorkflowId, UserId/Email, Role)
Koordinator -> Benutzerverwaltung	Existenz des Users/Emailadresse verifizieren Return: Username/Exception
Koordinator -> Workflow Configurator	addWorkflowUser(WorkflowId, UserId, Role)

Remove User from a Workflow Instance

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workflow Instance Configurator	Workflowteilnehmer entfernen
Browser/UI	Zeigt die Workflowteilnehmer samt Rollen an
Koordinator	Koordiniert Komponenten

1.2. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	removeWorkflowUser(WorkflowId, Userid)
Koordinator -> Workflow Configurator	removeWorkflowUser(WorkflowId, Userid)

Complete Work Item for a User

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workflow Instances Manager	Liste der Workflow-Instanzen zur Verfügung stellen
Human Task WS	Liste der Work items zur Verfügung stellen, Daten eines Work items zur Verfügung stellen und speichern
Data Storage WS	Evtl. hochgeladene Dateien speichern
Browser/UI	Zeigt die Workflow-Instanzen/WorkItems an
Koordinator	Koordiniert Komponenten
Session Manager	Verwaltet die aktuellen User-Sessions

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow Instances List	Workflow Instances Manager
Work items of a Workflow	Human Task WS
Work item Content	Human Task WS

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Work Item Formulareingaben (String)	Human Task (DB)	UI, Koordinator, Human Task
Hochgeladene Datei (binary)	Data Storage	UI, Koordinator, Data Storage

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Workflow-Instance-Name (String)	Workflow Instances Manager (DB)	Workflow Instances Manager, Koordinator, UI
Workitem-Name (String)	Human-Task-WS (DB)	Human-Task-WS, Koordinator, UI
Workitem-Content (String, DL-Link)	Human-Task-WS (DB)	Human-Task-WS, Koordinator, UI

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	getOwnWorkflowInstances(sessionID) return: Liste der Workflow Instances
Koordinator -> Session Manager	getUserIdFromSession(SessionID) return: UserID
Koordinator -> Workflow Instance Manager	getOwnWorkflowInstances(userID) return: Liste der Workflow Instances
UI -> Koordinator	getWorkflowWorkItems(sessionID, WorkflowInstanceID) return: Liste der Work items
Koordinator -> Human Task	Get WorkflowWorkItems(WorkflowInstanceID) return: Liste der Work items
UI -> Koordinator	getWorkItem(sessionID, WorkItemID) return: WorkItem Content
Koordinator -> Human Task WS	getWorkItem(WorkItemID) return: WorkItem Content
UI -> Koordinator	SubmitWorkItem(Eingegebene WorkItem-Daten)
Koordinator -> Human Task WS	SubmitWorkItem(Eingegebene WorkItem-Daten)
UI -> Data Storage	Upload von Binärdaten

Terminate Workflow Instance

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workflow Instances Manager	Workflow Instance stoppen
Browser/UI	Zeigt die Workflow-Instances an
Koordinator	Koordiniert Komponenten

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow Instances List	Workflow Instances Manager

1.3. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	terminateWorkflowInstance(WorkflowId, SessionID)
Koordinator -> Workflow Instance Manager	terminateWorkflowInstance(WorkflowId)

6.4.2 View Workflow Instances

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workflow-Instance-List	Enthält alle Workflow Instances des Workflow Admins.
Workflow-Instance-List-Seite	Zeigt Worfflow Instances des aktuellen Tanants des Workflow Admins an
Workflow-Instances-Managemenent System (WIMMS)	Verwaltet die Workflow Instances der Workflow Admins <ul style="list-style-type: none"> - beende Workflow-Instance - create Workflow-Instance - edit Workflow-Model Configuration
Benutzerverwaltung	Hole aktuellen Tenant-Namen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Worfflow Models des aktuellen Tanants des Tenant Admins.	Workflow Model List über WOMMS

1.3. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
Workflow-Instance-List-Seite, Workflow-Instance-Managemenent System (WIMMS)	Workflow-instance-seite holt die workflow instance liste über das workflow instance management system und zeigt diese an.
WIMMS, Benutzerverwaltung	WIMMS holt sich Tenant Name aus Benutzerverwaltung
WIMMS, Workflow-Instance-List	WIMMS holt Liste aus der Datenbank

6.4.4. Create Workflow Instance

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workflow-Model-Liste	Workflow Models anzeigen „Start Instance“ Button zur Verfügung stellen
Workflow Instance Configuration Interface	Workflow-Instanz-Variablen setzen/konfigurieren User zu Rollen hinzufügen Instanz-Name hinzufügen Start-Optionen setzen
Benutzerverwaltung	Prüfen, ob Workflow-Teilnehmer verfügbar sind Liste aller potentiell Workflow-teilnehmer zur Verfügung stellen
Email-Versand	Einladungen an Workflow-Teilnehmer versenden
Workflow-Engine	Führt den BPEL-Prozess aus
Transformationskomponente (eDWDL zu BPEL)	Transformiert eDWDL zu BPEL
WI-Configuration-Management	Nimmt vom Interface die Benutzereingaben entgegen und ruft Email-Versand, Datenbank-Zugriff, Benutzerverwaltung und Transformationskomponente auf und speichert die Workflow Configuration in eDWDL Registriert die angenommen Einladungen Lädt das Modell in DWDL und ermittelt die Konfigurations-Variablen
Datenbank-Zugriff	Ermöglicht Zugriff auf Models und Instance Configurations
Datenbank (Models und Instance-Configurations)	Speichert Models und Instance-Configurations
Workflow-Instance-Liste	Zeigt die laufenden und abgeschlossenen Workflow Instances an

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow Model Liste	Datenbank (DWDL) über die Datenbank-Zugriffskomponente
Zu konfigurierende Workflow Variablen und rollen	Datenbank (DWDL) über die Datenbank-Zugriffskomponente
Benutzerliste (Liste aller potentiellen Workflow-Teilnehmer)	Benutzerverwaltung

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Workflow-Variablen, entsprechend dem Workflow-Model	Datenbank (eDWDL)	Workflow Instance Configuration Interface WI-Configuration-Management Workflow-Instance-Liste Transformationskomponente (eDWDL zu BPEL)
User müssen sich registrieren (Boolean)	Datenbank (eDWDL)	WI-Configuration-Management
Auf Einladungen warten (Boolean)	Datenbank (eDWDL)	WI-Configuration-Management

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Workflow-Teilnehmer ist nicht verfügbar (Boolean)	Benutzerverwaltung	WI-Configuration-Management Workflow Instance Configuration Interface

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
Workflow-Model-Liste, Workflow Instance Configuration Interface (WICI)	Aufruf der Komponente WICI
Workflow-Model-Liste, Datenbank-Zugriff	Holen der Workflow Models
WI-Configuration-Management, Benutzerverwaltung	Ermittelt verfügbare Workflow-Teilnehmer
WI-Configuration-Management, Email-Versand	Verschicken von Einladungen
WI-Configuration-Management, Transformationskomponente (eDWDL zu BPEL)	Anweisung zur Transformation
WI-Configuration-Management, Datenbank (Models und Instance-Configurations)	Holt Modelle (DWDL) und speichert Konfigurationen (eDWDL)
Transformationskomponente (eDWDL zu BPEL), Workflow-Engine	Deployed den BPEL-Prozess
Workflow-Instance-Liste, Datenbank (Models und Instance-Configurations)	Holt die laufenden Instanzen und zeigt sie an

6.4.5 start workflow instance from a prepared one (siehe create from...)

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workflow-Instance-Liste	Zeigt alle Instancen, sowie die zugehörigen Daten an.

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow-Instance-Liste	Datenbank (eDWDL)

1.3. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
Workflow-Instance-Liste, Instance Configuration (WICI) Workflow Interface	Liste ruft WICI mit entsprechenden Übergabeparametern auf.

Define simple color scheme

Unklar: sollen die Daten aus der vorhanden CSS-Datei deserialisiert werden?

Vorschlag: UI speichert das geänderte CSS in eine Datei, die zu der entsprechenden TenantID zugeordnet ist, die CSS-Daten verlassen die UI also erst gar nicht

Wie die CSS-Daten aktualisiert werden, ist nicht klar!

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Koordinator	Koordiniert
Browser/UI	Nimmt die Änderungen entgegen und speichert die Tenant-CSS
Color Scheme -> CSS Serialisierer	Serialisiert die Einstellungen des Benutzers in eine CSS-Datei

Kommentar [Johannes 1]: Wird im Moment nicht gebraucht

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Vorheriges Farbschema	CSS-Datei innerhalb der UI

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Neues Farbschema (???)	Als CSS-Datei innerhalb de UI	UI, CSS-Serialisierer

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Neues Farbschema (Werte in Formularfeldern)	CSS-Datei innerhalb der UI	UI, CSS-(De)Serialisierer

1.4.Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> CSS-Serialisierer	GetCSSFormulardaten(TenantID) Return: CSS-Formulardaten
UI -> CSS-Serialisierer	generateCSS(TenantID, Formulardaten) return: success/Exception

Change tenant settings

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Koordinator	koordiniert
Tenant Management	Daten des Tenants bereitstellen und modifizieren
Browser/UI	Zeigt die Einstellungen an, nimmt Eingaben entgegen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Beschreibungstext	Tenant Management (Datenbank)

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Beschreibungstext (String)	Tenant Management (DB)	Browser/UI, Koordinator, Tenant Management

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Beschreibungstext (String)	Tenant Management (DB)	Browser/UI, Koordinator, Tenant Management

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
-------------	-----------------------

UI -> Koordinator	getTenantSettings(sessionID) Return: tenant settings/tenant not found exception
Koordinator -> Tenant Management	getTenantSettings(tenantID) Return: tenant settings/tenant not found exception
UI -> Koordinator	saveSettings(sessionID, newSettings) Return: current settings/tenant not found exception
Koordinator -> Tenant Management	saveSettings(tenantID, newSettings) Return: current settings/tenant not found exception

Kommentar [Johannes 1]: Wirklich nötig? Eigentlich reicht es ja, nach dem Save dann nochmal "getTenantSettings" aufzurufen...

View tenant members

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Koordinator	Koordiniert
Tenant Management	Stellt Benutzerliste bereit
Browser/UI	Zeigt Benutzerliste an

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Benutzerliste	Tenant Management (DB)

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Benutzerliste (Userlist)	Tenant Management (DB)	Tenant Management, Koordinator, UI

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	getUserList(sessionID) Return: Benutzerliste/tenant not found exception
Koordinator -> Tenant Management	getUserList(TenantID) Return: Benutzerliste/tenant not found exception

View other user's profile

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Koordinator	Koordiniert
Tenant Management	Stellt Benutzerliste bereit
Browser/UI	Zeigt Benutzerliste an
User Management	Stellt Benutzerdetails bereit

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Benutzerliste	Tenant Management (DB)
Benutzerdetails	User Management (DB)

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Benutzer (User)	keiner	User Management, Koordinator, UI

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Benutzerliste (Userlist)	Tenant Management (DB)	Tenant Management, Koordinator, UI

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
-------------	-----------------------

UI -> Koordinator	getUserList(sessionID) Return: Benutzerliste/tenant not found exception
Koordinator -> Tenant Management	getUserList(TenantID) Return: Benutzerliste/tenant not found exception
UI -> Koordinator	getUserDetails(sessionID, userID) Return: Details/user not found error
Koordinator -> User Management	getUserDetails(userID) Return: Details/user not found error

Create new workflow model

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Koordinator	Koordiniert
Browser/UI	Zeigt Daten an und nimmt Eingaben entgegen
Workflow Model Management	Legt Workflow Model an und zeigt bestehende Modelle an
Workflow Model Editor	Wird im UI angezeigt

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow Modelle	Workflow Model Management (DB)

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Modellname (String)	Workflow-Modell, DB	UI, Workflow Model Management

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Modellliste (Liste)	Workflow Model Management (DB)	UI, Workflow Model Management

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
-------------	-----------------------

UI -> Koordinator	getModelList(sessionID) Return: List<Models>/tenant not found exception
Koordinator -> Workflow Model Management	getModelList(TenantID) Return: List<Models>/tenant not found exception
UI -> Koordinator	createModel(sessionID, modelName) Return: modelEditor anzeigen/tenant not found exception/invalid model name
Koordinator -> Workflow Model Management	createModel(TenantID, modelName) Return: success/tenant not found exception/invalid model name
Koordinator -> Workflow Model Editor	editModel(TenantID, modelName) Return: modelEditor anzeigen/model not found exception

Kommentar [Johannes 1]: Falls die Validitätsprüfung rein syntaktisch ist, könnte sie direkt in der UI stattfinden

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Zeigt Modellliste an und nimmt Eingaben entgegen
Koordinator	Koordiniert
Workflow Model Management	Stellt Modellliste bereit und löscht Modelle

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Modellliste	Workflow Model Management (DB)

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Modellliste, löschen (Liste)	keiner	Workflow Model Management, Koordinator, UI

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Modellliste (Liste)	Workflow Model Management (DB)	Workflow Model Management, Koordinator, UI

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	getModelList(sessionID) Return: modellist/tenant not found exception
Koordinator -> Workflow Model	getModelList(TenantID)

Management	Return: modellist/tenant not found exception
UI -> Koordinator	deleteModels(sessionID, modelList) Return: success/tenant not found exception
Koordinator -> Workflow Model Management	deleteModels(sessionID, modelList) Return: success/tenant not found exception

6.7.10 Appoint Workflow Admin

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
WOMMS	<ul style="list-style-type: none"> - add workflow admin to workflow (email a) - add workflow admin to workflow (string a) - add workflow admin to workflow (user a) - hole workflow liste aus Datenbank
Benutzerverwaltung	<ul style="list-style-type: none"> - check if email in use - gebe emailadresse zu username - setze Workflow Admin-Rechte
Email Service	- sende notification an user
Appointment-Feld	Eingabe von Usernamen und dazugehörigen emailadressen
Workflow-List-Seite	Zeigt workflow Models des users an
Datenbank	Speichert Workflow-Liste
Benutzer-Pupup-Liste	Listet alle potentiellen Workflow Admins

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow-List	Datenbank (WOMMS)

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Username / Emailadresse / Boolean (just me)	Datenbank (Tabelle Modelliste)	<p>WOMMS</p> <p>Benutzerverwaltung</p> <p>Email Service</p> <p>Appointment-Feld</p>

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Username vorhanden (boolean)	Dynamisch ermittelt	Benutzer-Pupup-Liste Benutzerverwaltung

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
WOMMS, Benutzerverwaltung	<ul style="list-style-type: none"> - Prüfe ob User existiert - Setzt Workflow Admin Rechte - ermittle Email-Adressen von User
WOMMS, Email Service	- sende Einladung / Notification Email an künftigen Workflow Admin
Appointment-Feld, WOMMS	Weitergabe der Daten an WOMMS
Workflow-List-Seite, WOMMS	Hole Workflow Liste
WOMMS, Datenbank	Hole Workflow Liste aus der Datenbank
Benutzer-Pupup-Liste, Benutzerverwaltung	Hole passende Benutzer aus Benutzerverwaltung

Lock workflow models

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Zeigt workflow models an
Koordinator	Koordiniert als Hauptzweigstelle
Workflow-Model Manager	Greift auf die workflow models zu, um auf ihnen ein lock() auszuführen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow models	Workflow-Model Manager

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
LockingList(workflow IDs)	keiner	Workflow-Model Manager, UI, Koordinator

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
List(workflow IDs)	Workflow-Model Manager(DB)	Workflow-Model Manager, UI, Koordinator

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	Lock(sessionID, list)

		Return: list with locked workflow models
Koordinator Manager	-> Workflow-Model	Lock(list) Return: list with locked workflow models

Unlock workflow models

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Zeigt workflow models an und bietet create Buttons an
Koordinator	Koordiniert als Hauptzweigstelle
Workflow-Model Manager	Greift auf die workflow models zu, um auf ihnen ein unlock() auszuführen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow models	Workflow-Model Manager

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
UnlockingList(workflow IDs)		Workflow-Model Manager, UI, Koordinator

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
List(workflow IDs)	Workflow-Model Manager(DB)	Workflow-Model Manager, UI, Koordinator

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	Unlock(sessionID, list)

	Return: list with unlocked workflow models
Koordinator -> Workflow-Model Manager	unlock(list) Return: list with unlocked workflow models

Import published workflow model

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Zeigt published workflow models an und biete einen import button
Koordinator	Koordiniert als Hauptzweigstelle
Workflow-Model Manager	Greift auf die workflow models zu, um auf ihnen ein import() auszuführen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Published workflow models	Workflow-Model Manager

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
ImportList(workflow model IDs)		Workflow-Model Manager, UI, Koordinator

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
List(published workflow model IDs)	Workflow-Model Manager(DB)	Workflow-Model Manager, UI, Koordinator

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	import(sessionID, list) Return: success/exception
Koordinator -> Workflow-Model Manager	import(list) Return: success/exception

Publish workflow models

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Zeigt published workflow models an
Koordinator	Koordiniert als Hauptzweigstelle
Workflow-Model Manager	Greift auf die workflow models zu, um auf ihnen ein publish() auszuführen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow models	Workflow-Model Manager

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
PublishList(workflow model IDs)		Workflow-Model Manager, UI, Koordinator

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
List(workflow model IDs)	Workflow-Model Manager(DB)	Workflow-Model Manager, UI, Koordinator

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	publish(sessionID, list)

	Return: success/exception
Koordinator -> Workflow-Model Manager	publish(list) Return: success/exception

Unpublish workflow models

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Zeigt published workflow models an
Koordinator	Koordiniert als Hauptzweigstelle
Workflow-Model Manager	Greift auf die workflow models zu, um auf ihnen ein unpublish() auszuführen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow models	Workflow-Model Manager

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
UnpublishList(workflow model IDs)		Workflow-Model Manager

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
List(workflow model IDs)	Workflow-Model Manager(DB)	Workflow-Model Manager

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	unpublish(sessionID, list)

	Return: success/exception
Koordinator -> Workflow-Model Manager	unpublish (list) Return: success/exception

View list of tenant workflow models

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Zeigt workflow models an
Koordinator	Koordiniert als Hauptzweigstelle
Workflow-Model Manager	Greift auf die workflow models zu, um auf ihnen ein view() auszuführen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Tenant workflow models	Workflow-Model Manager

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
List(workflow model data)	Workflow-Model Manager(DB)	Workflow-Model Manager, UI, Koordinator

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	view(sessionID) Return: list of tenant workflow models
Koordinator -> Workflow-Model Manager	view(tenantID) Return: list of tenant workflow models

View list of published workflow models

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Browser/UI	Zeigt published workflow models an
Koordinator	Koordiniert als Hauptzweigstelle
Workflow-Model Manager	Greift auf die workflow models zu, um auf ihnen ein viewPublic() auszuführen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Published workflow models	Workflow-Model Manager

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
List(workflow model data)	Workflow-Model Manager(DB)	Workflow-Model Manager, UI, Koordinator

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
UI -> Koordinator	viewPublishedWorkflowModels(sessionID) Return: list with published workflow models
Koordinator -> Workflow-Model Manager	viewPublishedWorkflowModels() Return: list with published workflow models

6.8 Modelling Tool

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Toolbar	<ul style="list-style-type: none"> - insert new activity - insert new container - undo last action - redo last action
Property-Window	<ul style="list-style-type: none"> - edit container - edit activity
Zeichenebene	<ul style="list-style-type: none"> - clear Model - delete activity - delete container - connect container / activity
Variableneditor	<ul style="list-style-type: none"> - add variable - edit variable - delete variable
Rolleneditor	<ul style="list-style-type: none"> - add role - edit role - delete role
Formulardesigner	Konfiguriert Human task aktivität
DWDL Reader	<ul style="list-style-type: none"> - open model - convert DWDL to JS-Object
DWDL Writer	<ul style="list-style-type: none"> - save model - convert JS-Object to DWDL
Daten-Modell	Enthält, properties (Formulare für den Human Task), variablen, rollen
Grafik-Modell	Enthält grafische Elemente

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Workflow Model	Datenbank (DWDL)
Editierten Rollen,	DWDL Reader (liest DWDL)

Variablen, Formulare	
----------------------	--

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Dateityp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Eingabe:

Datum (Typ)	Speicherort	Komponenten
Model (aus grafischer Erstellung)	Datenbank (DWDL)	<ul style="list-style-type: none"> - Transformationskomponente zu BPEL - Workflow-Instance Konfigurator
Variablen, Rollen, Formulare, Properties	Datenbank (DWDL)	<ul style="list-style-type: none"> - Transformationskomponente zu BPEL - Workflow-Instance Konfigurator

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
DWDL	Datenbank(DWDL)	<ul style="list-style-type: none"> - DWDL Reader - DWDL Writer
DWDL (grafische Darstellung)	-	Zeichenebene Variableneditor Rolleneditor Property-Window Formular designer

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
DWDL Reader, Datenmodell	Liest properties (Formulare für den Human Task), variablen, rollen aus dem DWDL aus und speichert diese im Datenmodell
DWDL Reader, Grafikmodell	Liest Grafikobjekte aus dem DWDL aus und speichert diese im Datenmodell.
DWDL Writer, Datenmodell/Grafikmodell	Liest Datenmodell und Grafikmodell und speichert Informationen im DWDL.

Property-Window, Datenmodell	<ul style="list-style-type: none">- Liest Properties aus Datenmodell aus- Speichert Properties in Datenmodell
Zeichenebene, Grafikmodell	<ul style="list-style-type: none">- Liest grafische Objekte aus Grafikmodell aus- Speichert grafische Objekte in Grafikmodell
Variableneditor, Datenmodell	<ul style="list-style-type: none">- Liest Variablen aus Datenmodell aus- Speichert Variablen in Datenmodell
Rolleneditor, Datenmodell	<ul style="list-style-type: none">- Liest Rollen aus Datenmodell aus- Speichert Rollen in Datenmodell
Formulardesigner, Datenmodell	<ul style="list-style-type: none">- Liest Formulare aus Datenmodell aus- Speichert Formulare in Datenmodell
DWDL Writer / WOMMS	DWDL Writer übergibt DWDL an WOMMS.

6.3.5 Mark work item as done

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workitem-List-Seite	Zeigt Workitem-List an
Work-Item-List	Liste der Workitems mit den Items und Eigenschaften. WIMS greift auf diese zu.
Workitem-Management-System (WIMS)	Verwaltet die Work-Items. <ul style="list-style-type: none"> - erstelle workitem - hole workitem - setze workitem status - lösche Workitem
Workflow Engine	Nimmt Aufruf des HAT-Webservices entgegen, setzt gegebenenfalls den Workflow fort.
Human-Task-Webservice	Beendet Workitem und ruft Workflow Engine auf.

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Work-Item-List	Workitem-Management-System

1.3. Welche Eingabe- / Ausgabedaten bezogen auf den Use-Case gibt es? Von welchem Datentyp sind sie? Wo werden sie gespeichert? Von welchen anderen Komponenten werden sie eventuell verwendet?

Ausgabe:

Datum (Typ)	Speicherort	Komponenten
Work Item Objekt		HT-Webservice

1.4. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
Workitem-Management-System (WIMS), Human-Task-Webservice	Weitergabe der Daten aus der Datenbank an den Human Tak Webservice.
Human-Task-Webservice , Workflow Engine	Asynchroner Invoke an Workflow Engine.
Workitem-List-Seite, Workitem-Management-System	Abfrage der Workitem-List des Benutzers beim WIMS
Workitem-Management-System , Work-Item-List	WIMS holt die Workitemlist aus der DB

6.4.1 View Workflow Models

1.1. Welche (System)Komponenten sind an dem Use-Case beteiligt? Welche Funktionalität müssen diese für den Use-Case bereitstellen?

Komponente	Funktionalität
Workflow-Model-List	Enthält alle Workflow Models des Tenant Admins.
Workflow-Model-List-Seite	Zeigt Worfflow Models des aktuellen Tanants des Tenant Admins an
Workflow-Model-Managemenent System (WOMMS)	Verwaltet die Workflow Models der Tenant Admins <ul style="list-style-type: none"> - lösche Model - update Model - publish Model - import model - create model - set workflow admins - set model locked / unlocked
Benutzerverwaltung	Hole aktuellen Tenant-Namen

1.2. Welche Daten werden angezeigt (falls welche angezeigt werden)? Aus welcher Quelle kommen diese Daten?

Datum	Quelle
Worfflow Models des aktuellen Tanants des Tenant Admins.	Workflow Model List über WOMMS

1.3. Zwischen welchen Komponenten findet intern eine Kommunikation statt?

Komponenten	Art der Kommunikation
Workflow-Model-List-Seite, Workflow-Model-Management System (WOMMS)	WMLS holt WML von WOMMS und zeigt diese an.
Workflow-Model-Managemenent System (WOMMS), Benutzerverwaltung	WOMMS holt sich Tenant Name aus Benutzerverwaltung
WOMMS, Workflow-Model-List	WOMMS holt Liste aus der Datenbank