Pflichtenheft
SWARM Composer
-
Softwareprojekt SoSe18
Gruppe HRS 3 105b

Jeremia Böhmig
Felix Gröner
Janek Haberer
Robert Köhler
Johanna Menzel
Jette Petzold
Christian Richter
Connor Schönberner

se

2. September 2018

Inhaltsverzeichnis

Einleitung

Dies ist das Pflichtenheft des SWARM Composers. Diese Software ist Teil des Forschungsprojektes SWARM des Unternehmens adesso. Der SWARM Composer dient dazu, unterschiedliche Software für Bauprojekte in Bezug auf ihre Kompatibilität bezüglich der Ein- und Ausgabeformate zu überprüfen. Der SWARM Composer besteht aus zwei Teilen: einem Webserver und einer App. Auf dem Webserver können Benutzer und Benutzerinnen Dienste zu Kompositionen zusammenfassen und auf Kompatibilität überprüfen. Neue Dienste können dabei manuell eingegeben oder durch eine JSON-Datei eingelesen werden. Dies ist aber nur mit Administratorrechten möglich. In der App können Kompositionen grafisch präsentiert und per PDF verschickt werden.
Unter einem **Dienst** wird ein Programm verstanden das Daten verarbeitet und das mindestens entweder ein Eingabe- oder Ausgabeformat besitzt.
Eine **Komposition** ist eine Zusammenfassung von Diensten mit der Information, welcher Dienst welchem anderen welche Daten sendet, und deren Kompatibilitätsbeziehungen.
Zwei Dienste sind **kompatibel**, wenn das Eingabeformat des empfangenden Dienstes mit dem Ausgabeformat des sendenden Dienstes übereinstimmt.
Wenn im Folgenden von **Bearbeiten** die Rede ist, so beinhaltet dies stets auch das **Löschen**.

Kapitel 1

Lizenz

Kapitel 2

Zielbestimmungen

Im Folgenden ist der Funktionsumfang des SWARM Composers, insbesondere der Web-App und der Android-App, aufgeführt. Dabei werden funktionale und nicht-funktionale Zielbestimmungen getrennt aufgeführt. Des Weiteren wird zwischen Muss-, Soll-, Kann- und Abgrenzungskriterien unterschieden.

- Musskriterien umfassen alle Ziele und Funktionalitäten, die für einen Einsatz des entwickelten Produktes unabdingbar sind. Sie müssen daher ohne Kompromisse implementiert werden. Ein Wegfall eines einzelnen Musskriteriums würde das Produkt außer Betrieb setzen.
- Sollkriterien sind gewünschte Funktionen, die ebenfalls implementiert werden müssen, deren Wegfall auf Grund von unüblichen Umständen jedoch nicht den Einsatz des Produkts verhindern würde.
- Kannkriterien sind alle Ziele, die wünschenswert sind, aber nicht zwingend notwendige Funktionen darstellen.
- Abgrenzungskriterien zeigen, was explizit **nicht** umgesetzt wird. Sie dienen dazu, die Grenzen des Produkts zu definieren.

Funktionale Zielbestimmungen

Musskriterien

- Der SWARM Composer muss über eine native Android-App erreichbar sein.
- Der SWARM Composer muss über eine Website erreichbar sein.
- Neue Benutzende des SWARM Composers müssen sich über die Website registrieren können.
- Registrierte Benutzende des SWARM Composers müssen sich anmelden können, unabhängig davon, ob sie über die Website oder die Android-App zugreifen.
- Administrierende müssen eine JSON-Datei hochladen können, um neue Dienste zu erstellen.

- Administrierende müssen über eine Webmaske neue Dienste hinzufügen können.
- Beim manuellen Erstellen von Diensten über eine Webmaske muss das Format syntaktisch überprüft werden.
- Administrierende müssen die Möglichkeit haben, Dienste zu überarbeiten und zu löschen.
- Auf der Website muss die Möglichkeit bestehen, Kompositionen zu erstellen.
- Der SWARM Composer muss eine Kompatibilitätsprüfung für Kompositionen durchführen.
- Es muss eine grafische Rückmeldung über die Kompatibilität von Diensten erfolgen.
- Die Speicherung von erstellten Kompositionen muss möglich sein.
- Die Android-App muss erstellte Kompositionen anzeigen können.

Sollkriterien

- Unter allen Nutzenden soll zwischen den zwei Rollen normalem Nutzenden und Administrierenden unterschieden werden, wobei Letztere über mehr Nutzungsrechte verfügt.
- Bestehende Kompositionen sollen bearbeitet werden können.
- Der Erstellende einer Komposition soll festlegen können, welche Nutzenden diese einsehen dürfen.
- Öffentliche Kompositionen sollen auch ohne Benutzerkonto sichtbar sein.
- Wer eine Komposition erstellt, soll festlegen können, welche Nutzenden diese verändern können.
- In der Android-App soll die Möglichkeit bestehen, Kompositionen als PDF zu teilen.
- Es sollen Konverter angezeigt werden, wenn Dienste nicht kompatibel sind.
- Bei der manuellen Eingabe von Diensten sollen Schlagwörter vorgeschlagen werden.
- Beim Erstellen von Kompositionen können Dienste nach Namen und Schlagwörter gefiltert werden.

Kannkriterien

- Die Android-App kann zu einzelnen Diensten die gespeicherten Informationen detailliert anzeigen.
- Es gibt eine Suchfunktion für Kompositionen.
- Auf der Website können Kompositionen verschickt werden.

Abgrenzungskriterien

- In der Android-App können keine Kompositionen erstellt werden.
- In der Android-App können keine Dienste erstellt werden.
- In der Android-App können keine Benutzenden registriert werden.

- Beim manuellen Erstellen von Diensten wird nicht geprüft, ob der Inhalt korrekt ist.
- Es gibt keine benutzerspezifischen Einschränkungen bei der Sichtbarkeit einzelner Dienste.
- Kompositionen können nicht aus Kompositionen zusammengesetzt werden

Nicht funktionale Zielbestimmungen

Musskriterien

- Die Web-Applikation muss mit Java-Spring erstellt werden.
- Die Android-App muss ab der Android Version 6 benutzbar sein.
- Die verwendeten URLs müssen zur Laufzeit änderbar sein.

Sollkriterien

- Die Kommunikation zwischen Web-App und Android-App soll verschlüsselt sein.
- Es soll eine Drag-and-Drop-Benutzeroberfläche auf der Website geben.
- Die Benutzeroberfläche soll einfach und intuitiv gestaltet sein.
- Der Quellcode des SWARM Composers soll unter eine geeignete Lizenz gestellt werden.

Kapitel 3

Produkteinsatz

Im Folgenden wird kurz darauf eingegangen, in welchem Kontext, von wem und unter welchen Bedingungen der SWARM Composer eingesetzt wird.

Anwendungsgebiete

Der SWARM Composer dient Anwendern und Anwenderinnen des SWARM-Ökosystems dazu, Kompositionen von Diensten auf ihre Kompatibilität zu prüfen. Über das Ergebnis dieser Prüfung gibt das System grafisch eine Rückmeldung. Gleichzeitig lässt sich mit ihm ein Überblick über die vorhandenen atomaren Dienste erlangen sowie Kompositionen verwalten.

Zielgruppen

Primär richtet sich der SWARM Composer an Benutzende aus der Architektur- und Baubranche, ohne dass weiteres Zusatzwissen über die interne Funktionsweise des Systems vorausgesetzt wird. Allerdings ist Wissen über übliche Services, die im SWARM-Ökosystem genutzt werden, hilfreich, um Kompositionen zu erstellen.
Es sind zwei Rollen vorgesehen: Zum einen Standard-Nutzende, die sowohl Dienste zu Kompositionen zusammensetzen als auch diese prüfen, speichern und teilen können, zum anderen Administrierende, die zusätzlich Dienste anlegen und verwalten können. Hierfür muss das von Adesso spezifizierte Format für Dienste bekannt sein.

Betriebsumgebungen

Zum Betrieb des Systems ist ein Server-Computer, auf dem die Webanwendung laufen kann, zwingend notwendig. Wenn die Nutzung der App gewünscht ist, ist hierfür ein Android-Mobilgerät in Form eines Tablets oder Smartphones notwendig. Für die auf einem Server (beispielsweise in Form von Apache Tomcat) laufende Anwendung ist ganztägige Uptime und autonomes Laufen gewährleistet. Es findet eine automatische Datensicherung statt.

Kapitel 4

Produktumgebung

Um die Software fehlerfrei und korrekt zu nutzen, müssen bestimmte technische Bedingungen erfüllt sein, die hier aufgeführt werden.

Software

Für das Webinterface dient ein einfacher Webbrowser als Client. Um die App zu benutzen, wird Android 6 oder neuer vorausgesetzt.

Auf dem Server muss eine Linux-Distribution mit Docker und Jenkins installiert sein.

Hardware

Für die App muss das Gerät Android 6 oder neuer unterstützen.

Webservices müssen auf dem Server laufen können.

Orgware

Sowohl Server als auch die Clients müssen netzwerkfähig sein.

Produktschnittstelle

Die Kommunikation zwischen Server und Client findet bei dem Webinterface über HTTPS und bei der App über eine REST-Schnittstelle statt. Für das Verschicken von exportierten PDFs wird die Standard Android API verwendet.

Kapitel 5

Produktfunktionen

Die Produktfunktionen beschreiben jede einzelne Funktion des Produkts mittels Anwendungsfalldiagrammen und Anwendungsfalltabellen.

In Tabelle Abbildung 5.1 werden alle auftretenden Akteure beschrieben.

Akteur	**Beschreibung**	**Verwendet in Anwendungsszenario**
unregistrierter Nutzer	Nicht angemeldeter Nutzer: Kann nur Kompositionen einsehen	APP-2, APP-3, WEB-6
Nutzer	Angemeldeter Nutzer: Nutzt den SWARM Composer	APP-1, APP-2, APP-3, APP-4, WEB-4, WEB-5, WEB-6
Administrator	Nutzt und verwaltet den SWARM Composer	APP-1, APP-2, APP-3, APP-4, WEB-1, WEB-2, WEB-3, WEB-4, WEB-5, WEB-6

Abbildung 5.1: Beschreibung der Akteure

5.1 Anwendungsfalldiagramm - App

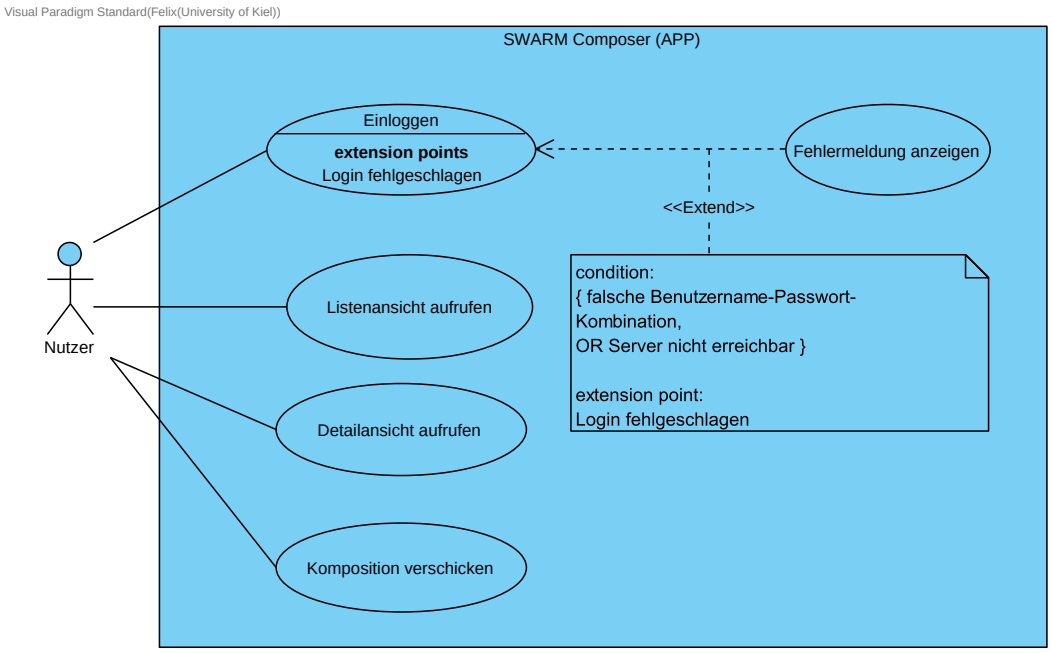


Abbildung 5.2: Anwendungsfalldiagramm - App

Anwendungsfall ID	APP-1
Anwendungsfallname	Einloggen
Initiierender Akteur	Nutzer
Weitere Akteure	-
Kurzbeschreibung	Ein Nutzer loggt sich ein
Vorbedingungen	Android-App ist geöffnet, der Nutzer ist nicht eingeloggt
Nachbedingungen	Der Nutzer ist eingeloggt
Ablauf	1. Der Nutzer gibt seinen Benutzernamen ein 2. Der Nutzer gibt sein Passwort ein 3. Der Nutzer tippt auf den Login-Button 4. Bestätigung der erfolgreichen Anmeldung wird angezeigt
Alternative	-
Ausnahme	1. Der Nutzer gibt den Benutzernamen ein 2. Der Nutzer gibt das Passwort ein 3. Der Nutzer tippt auf den Login-Button 4. Anmeldung schlägt fehl und eine Fehlermeldung wird angezeigt
Benutzte Anwendungsfälle	Fehlermeldung anzeigen
Spezielle Anforderungen	-
Annahmen	-

Abbildung 5.3: Anwendungsfall APP-1

Anwendungsfall ID	APP-2
Anwendungsfallname	Listenansicht aufrufen
Initiierender Akteur	Nutzer
Weitere Akteure	-
Kurzbeschreibung	Dem Nutzer wird eine Liste von vorhandenen Kompositionen angezeigt
Vorbedingungen	Android-App ist geöffnet
Nachbedingungen	Dem Nutzer wird eine Liste von vorhandenen Kompositionen angezeigt
Ablauf	1. Die für den Nutzer sichtbaren Kompositionen werden vom Server abgerufen 2. Daten werden als Liste angezeigt
Alternative	1. Der Nutzer kehrt zur Listenansicht zurück 2. Dem Nutzer wird eine Liste der sichtbaren Kompositionen angezeigt
Ausnahme	-
Benutzte Anwendungsfälle	-
Spezielle Anforderungen	-
Annahmen	-

Abbildung 5.4: Anwendungsfall APP-2

Anwendungsfall ID	APP-3
Anwendungsfallname	Detailansicht aufrufen
Initiierender Akteur	Nutzer
Weitere Akteure	-
Kurzbeschreibung	Nutzer ruft die Detailansicht einer Komposition auf
Vorbedingungen	Nutzer hat die Listenansicht aufgerufen
Nachbedingungen	Nutzer wird die Detailansicht einer Komposition gezeigt
Ablauf	1. Nutzer wählt in der Listenansicht eine Komposition aus 2. Nutzer werden die Details der ausgewählten Komposition angezeigt
Alternative	-
Ausnahme	-
Benutzte Anwendungsfälle	-
Spezielle Anforderungen	-
Annahmen	-

Abbildung 5.5: Anwendungsfall APP-3

Anwendungsfall ID	APP-4
Anwendungsfallname	Komposition verschicken
Initiierender Akteur	Nutzer
Weitere Akteure	-
Kurzbeschreibung	Nutzer verschickt eine Komposition
Vorbedingungen	Nutzer hat in der Listenansicht eine Komposition ausgewählt und die Detailansicht aufgerufen
Nachbedingungen	Komposition wird verschickt
Ablauf	1. Nutzer wird eine Komposition in der Detailansicht angezeigt 2. Nutzer tippt auf den Verschicken-Button 3. Eine Datei im PDF-Format wird erstellt 4. Nutzer bestätigt den Versand
Alternative	-
Ausnahme	-
Benutzte Anwendungsfälle	-
Spezielle Anforderungen	-
Annahmen	-

Abbildung 5.6: Anwendungsfall APP-4

5.2 Anwendungsfalldiagramm - Server

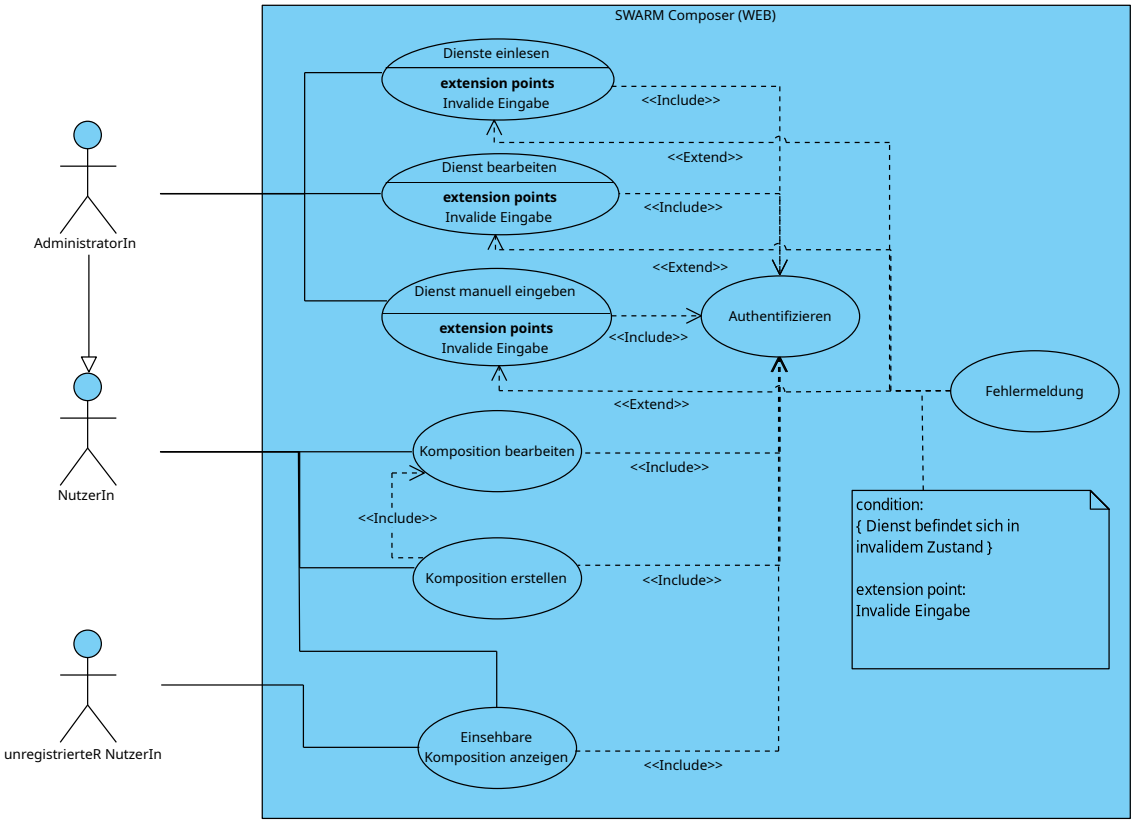


Abbildung 5.7: Anwendungsfalldiagramm - Server

Anwendungsfall ID	WEB-1
Anwendungsfallname	Dienst manuell einfügen
Initiierender Akteur	Administrator
Weitere Akteure	-
Kurzbeschreibung	Neuer Dienst wird in die Datenbank eingefügt
Vorbedingungen	Administrator ist eingeloggt und befindet sich auf Administrationsseite
Nachbedingungen	Neuer Dienst wurde in Datenbank gespeichert
Ablauf	1. [Use-Case: Authentifizieren] 2. Administrator wählt "Dienst eingeben" aus 3. Administrator gibt Dienstdetails in Eingabemaske ein 4. Administrator bestätigt die Eingabe 5. System akzeptiert Eingabe und speichert in Datenbank 6. Weiterleitung zur Administrationsseite
Alternative	-
Ausnahme	1. [Use-Case: Authentifizieren] 2. Administrator wählt "Dienst eingeben" aus 3. Administrator gibt Dienstdetails in Eingabemaske ein 4. Administrator bestätigt die Eingabe 5. System akzeptiert Eingabe nicht, da sie syntaktisch ungültig ist 6. System zeigt Fehler an und markiert ungültig Felder 7. Es wird in der Eingabemaske verblieben
Benutzte Anwendungsfälle	Authentifizieren
Spezielle Anforderungen	-
Annahmen	Authentifizierung ist erfolgreich

Abbildung 5.8: Anwendungsfall WEB-1

Anwendungsfall ID	WEB-2
Anwendungsfallname	Dienste einlesen
Initiierender Akteur	Administrator
Weitere Akteure	-
Kurzbeschreibung	Ein oder mehrere Dienste werden über eine JSON Datei eingelesen, die sich lokal auf einem Nutzerrechner befindet
Vorbedingungen	Administrator ist eingeloggt und befindet sich auf der Administrationsseite
Nachbedingungen	Neue Dienste wurden der Datenbank hinzugefügt
Ablauf	1. [Use-Case: Authentifizieren] 2. Administrator wählt "Dienste einlesen" aus 3. Administrator wählt die hochzuladene JSON-Datei aus dem sich öffnenden Dateibrowser vom lokalen Rechner aus 4. Eingelesene Daten werden vom System akzeptiert und aufgelistet 5. Administrator speichert 6. Dienste werden in der Datenbank gespeichert 7. Weiterleitung auf Administrationsseite
Alternative	-
Ausnahme	1. [Use-Case: Authentifizieren] 2. Administrator wählt "Dienste einlesen" aus 3. Administrator wählt hochzuladende JSON Datei aus dem sich öffnenden Dateibrowser vom lokalen Rechner aus 4. System erkennt Fehler in der JSON-Datei und gibt einen Fehler aus 5. Administrator bleibt auf der Administrationseite
Benutzte Anwendungsfälle	Authentifizieren
Spezielle Anforderungen	-
Annahmen	Authentifizierung ist erfolgreich

Abbildung 5.9: Anwendungsfall WEB-2

Anwendungsfall ID	WEB-3
Anwendungsfallname	Dienst bearbeiten
Initiierender Akteur	Administrator
Weitere Akteure	-
Kurzbeschreibung	Felder eines existierenden Dienstes werden bearbeitet
Vorbedingungen	Administrator ist eingeloggt und befindet sich auf Administrationsseite
Nachbedingungen	Vorgenommene Änderungen am Dienst wurden in Datenbank übernommen
Ablauf	1. [Use-Case: Authentifizieren] 2. Administrator wählt Dienst aus Liste von vorhandenen Diensten aus 3. Administrator wählt “Bearbeiten” 4. Administrator führt Änderungen in Eingabemaske durch 5. Administrator bestätigt die Eingabe 6. System akzeptiert die Änderung und speichert diese in der Datenbank 7. Weiterleitung zur Administrationsseite
Alternative	-
Ausnahme	1. [Use-Case: Authentifizieren] 2. Administrator wählt Dienst aus Liste von vorhandenen Diensten aus 3. Administrator wählt “Bearbeiten” 4. Administrator tätigt eine invalide Eingabe 5. Administrator bestätigt die Eingabe 6. System erkennt Fehler und zeigt eine Fehlermeldung an 7. Administrator bleibt auf der Administrationsseite
Benutzte Anwendungsfälle	Authentifizieren
Spezielle Anforderungen	-
Annahmen	Authentifizierung ist erfolgreich

Abbildung 5.10: Anwendungsfall WEB-3

Anwendungsfall ID	WEB-4
Anwendungsfallname	Komposition erstellen
Initiierender Akteur	Nutzer
Weitere Akteure	-
Kurzbeschreibung	Eine neue Komposition wird zum System hinzugefügt
Vorbedingungen	Nutzer befindet sich auf der Übersichtsseite und ist eingeloggt
Nachbedingungen	Komposition ist erstellt und Nutzer befindet sich im Bearbeitungsmodus
Ablauf	1. [Use-Case: Authentifizieren] 2. Nutzer wählt "Komposition erstellen" aus 3. Nutzer gibt einen Namen für die Komposition an 4. Nutzer wird zur Bearbeitungsseite weitergeleitet 5. [Use-Case: Komposition bearbeiten]
Alternative	-
Ausnahme	-
Benutzte Anwendungsfälle	WEB-5, Authentifizieren
Spezielle Anforderungen	-
Annahmen	Authentifizierung ist erfolgreich

Abbildung 5.11: Anwendungsfall WEB-4

Anwendungsfall ID	WEB-5
Anwendungsfallname	Komposition bearbeiten
Initiierender Akteur	Nutzer
Weitere Akteure	-
Kurzbeschreibung	Nutzer bearbeitet Komposition
Vorbedingungen	Nutzer besitzt benötigte Rechte zur Bearbeitung der Komposition und befindet sich auf der Übersichtsseite
Nachbedingungen	Nutzer befindet sich auf der Bearbeitungsseite
Ablauf	1. [Use-Case: Authentifizieren] 2. Nutzer wählt “Komposition bearbeiten” aus 3. Nutzer wird zur Bearbeitungsseite weitergeleitet 4. Nutzer wird in den Bearbeitungsmodus versetzt
Alternative	-
Ausnahme	-
Benutzte Anwendungsfälle	Authentifizieren
Spezielle Anforderungen	-
Annahmen	Es werden nur Kompositionen gezeigt, für die der Nutzer die benötigten Rechte besitzt Der Bearbeitungsbutton ist ausgegraut, falls die Bearbeitungsrechte nicht vorhanden sind. Die Authentifizierung war erfolgreich.

Abbildung 5.12: Anwendungsfall WEB-5

Anwendungsfall ID	WEB-6
Anwendungsfallname	Einsehbare Komposition anzeigen
Initiierender Akteur	unregistrierter Nutzer oder Nutzer
Weitere Akteure	-
Kurzbeschreibung	(Unregistrierter) Nutzer lässt sich eine Komposition anzeigen
Vorbedingungen	(Unregistrierter) Nutzer befindet sich auf der Übersichtsseite
Nachbedingungen	(Unregistrierter) Nutzer wird die Komposition angezeigt
Ablauf	1. [Use-Case: Authentifizieren] 2. (Unregistrierter) Nutzer wählt die anzuzeigende Komposition aus 3. Dem (unregistrierten) Nutzer wird die Komposition angezeigt
Alternative	-
Ausnahme	-
Benutzte Anwendungsfälle	Authentifizieren
Spezielle Anforderungen	-
Annahmen	Es werden nur Kompositionen gezeigt, für die der (unregistrierte) Nutzer die benötigten Rechte besitzt. Die Authentifizierung war erfolgreich.

Abbildung 5.13: Anwendungsfall WEB-6

Kapitel 6

Testfälle

In folgendem Kapitel werden verschiedene Benutzungsszenarien der Software beschrieben und das erwartete Verhalten definiert. In Tabelle 6.1 wird das Verhalten der Android-App definiert, in Tabelle 6.2 und Tabelle 6.3 das der Web-App.

Nr.	Anwendungsfall ID	Szenario	Erwartetes Verhalten
1	App-1	Beim Einloggen wird ein nicht registrierter Nutzername eingegeben.	Es wird eine passende Fehlermeldung angezeigt, ein Login kann erneut versucht werden.
2	App-1	Das zu einem registrierten Nutzernamen eingegebene Passwort ist nicht korrekt.	Es wird eine passende Fehlermeldung angezeigt, ein Login kann erneut versucht werden.
3	App-1	Während des Loginvorgangs ist der Server nicht mehr erreichbar.	Eine passende Fehlermeldung wird angezeigt.
4	App-1	Ein registrierter Nutzername und das dazu passende Passwort werden eingegeben.	Der Login ist erfolgreich und es wird die erweiterte Listenansicht der Kompositionen angezeigt.
5	App-2	Die Listenansicht wird von einer eingeloggten Person angefordert.	Eine Liste der für diese Person einsehbaren Kompositionen wird angezeigt.
6	App-2	Die Listenansicht wird von einer nicht eingeloggten Person angefordert.	Es erscheint eine Listenansicht der öffentlichen Kompositionen.
7	App-3	Es wird die Detailansicht einer Komposition angefordert.	Die Komposition wird mit den Kompatibilitätshinweisen zwischen den einzelnen Diensten graphisch angezeigt.
8	App-4	Das Versenden einer Komposition als PDF schlägt fehl.	Es wird eine passende Fehlermeldung angezeigt.
9	App-4	Das Versenden einer Komposition als PDF war erfolgreich.	Es wird eine Bestätigung angezeigt.

Abbildung 6.1: Beschreibung der Testfälle für die Android-App

Nr.	Anwendungsfall ID	Szenario	Erwartetes Verhalten
1	Web-1/2	Nutzende ohne Administrator-Rechte versuchen, einen Dienst einzufügen.	Die Aktion wird verweigert und es wird eine passende Fehlermeldung angezeigt.
2	Web-1	Nutzende mit Administrator-Rechten wählen aus, einen Dienst manuell einzufügen.	Es wird das Eingabeformular für Dienste angezeigt.
3	Web-1	Das Eingabeformular wird syntaktisch korrekt abgeschickt.	Der neue Dienst wird in die Datenbank einfügt.
4	Web-1	Beim Eingabeformular wird das Schlagwortfeld ausgewählt.	Es werden zur Verfügung stehende Schlagwörter angezeigt.
5	Web-1	Das Eingabeformular wird nicht syntaktisch korrekt abgeschickt.	Es wird ein Hinweis angezeigt.
6	Web-2	Eine Datei im passenden JSON-Format wird eingelesen.	Die Dienste der Datei werden in die Datenbank aufgenommen und es wird eine Erfolgsmeldung angezeigt.
7	Web-2	Eine Datei in syntaktisch nicht passenden JSON-Format wird eingelesen.	Es wird eine entsprechende Fehlermeldung angezeigt.
8	Web-3	Nutzende ohne Administrator-Rechte versuchen einen Dienst zu bearbeiten.	Es wird eine passende Fehlermeldung angezeigt.
9	Web-3	Ein Dienst wird bearbeitet und dadurch invalide.	Die Änderung wird verweigert.
10	Web-3	Ein Dienst wird korrekt bearbeitet.	Die Änderung wird in die Datenbank übernommen und es erscheint eine Erfolgsnachricht.
11	Web-4	Eine unregistrierte Person möchte eine Komposition erstellen.	Die Aktion wird mit Fehlermeldung verweigert.
12	Web-4	Eine registrierte Person möchte eine Komposition erstellen.	Es wird die Weboberfläche zum Bearbeiten einer Komposition angezeigt.
13	Web-4	Eine erstellte Komposition soll gespeichert werden.	Die Komposition wird mit den angegebenen Einstellungen in die Datenbank übernommen.

Abbildung 6.2: Beschreibung der Testfälle für die Web-App

Nr.	Anwendungsfall ID	Szenario	Erwartetes Verhalten
14	Web-5	Eine Person mit nicht ausreichenden Rechten versucht, eine Komposition zu bearbeiten.	Die Aktion wird verweigert und es wird eine passende Fehlermeldung ausgegeben.
15	Web-5	Eine Person mit ausreichenden Rechten möchte eine Komposition bearbeiten.	Die Weboberfläche zum Bearbeiten einer Komposition wird angezeigt.
16	Web-5	Eine bearbeitete Komposition soll gespeichert werden.	Die Änderungen werden in die Datenbank übernommen.
17	Web-6	Eine Person möchte eine für sie nicht einsehbare Komposition einsehen.	Es wird eine entsprechende Fehlermeldung angezeigt und die Aktion verweigert.
18	Web-6	Eine Person möchte eine für sie einsehbare Komposition einsehen.	Die gewählte Komposition wird visuell dargestellt.

Abbildung 6.3: Beschreibung der Testfälle für die Web-App

Kapitel 7

Produktdaten

Um die Funktionsweise des Dienstes zu realisieren, ist es notwendig, die hier genannten Daten zu speichern.
Zu den Benutzenden werden folgende Daten gespeichert:

- Anrede
- Name
- E-Mail-Adresse
- Passwort
- Erstellte Kompositionen
- Kompositionen, die eingesehen oder geändert werden dürfen
- vorhandene Administratorrechte

Zu den Diensten werden folgende Daten gespeichert:

- Name
- Organisation
- Version
- Schlagworte
- gültige Eingabeformate (ggf. auch keine)
- gültige Ausgabeformate (ggf. auch keine)

Zu den Kompositionen werden folgende Daten gespeichert:

- Autor
- Enthaltene Dienste
- Kompatibilitätsbeziehungen der Dienste
- Sichtbarkeit (öffentlich, nur bestimmte Benutzer)

- Zum Ändern berechtigte Benutzende

Zu den Ein- und Ausgabeformaten werden folgende Daten gespeichert:

- Name des Eingabe- und Ausgabeformats (z.B JSON)
- Version
- Abwärtskompatibilität

Bei der Abwärtskompatibilität wird zwischen *strict* und *flexible* unterschieden: Mit *flexible* markierte Formate sind abwärtskompatibel, mit *strict* gekennzeichnete sind dies hingegen nicht.

Kapitel 8

Benutzeroberfläche

In diesem Kapitel werden erste Entwürfe der Benutzeroberflächen dargestellt. Dabei sind weder die ästhetischen Aspekte noch die Anordnung der Inhalte in ihrer finalen Version.

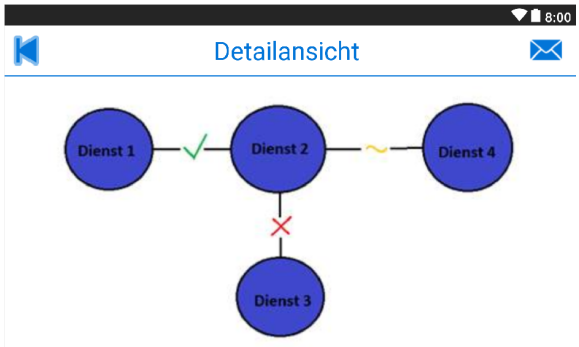
Skizze	Beschreibung
	In der Listenansicht werden den Nutzenden alle für sie sichtbaren Kompositionen mit einigen Informationen (Urheber, Erstellungsdatum, ...) angezeigt. Wer nicht eingeloggt ist, sieht nur die öffentlichen Kompositionen. Durch ein Tippen auf das Lupen-Symbol besteht die Möglichkeit, die Liste zu filtern. Durch das Antippen des Zahnrad-Symbols gelangen die Nutzenden in ein Menü, in dem sie sich unter anderem ein- und ausloggen können.
	Durch das Antippen einer Komposition aus der Listenansicht wird die Detailansicht geöffnet. Hier wird die Komposition als Graph dargestellt. (Fehlende) Kompatibilität wird an den Verbindungen gekennzeichnet. Mit dem Brief-Symbol kann die angezeigte Komposition verschickt werden. Dabei wird ein PDF erstellt, das sowohl den Graphen als auch einige Details in textueller Form enthält.

Abbildung 8.1: Mockup der Android-App

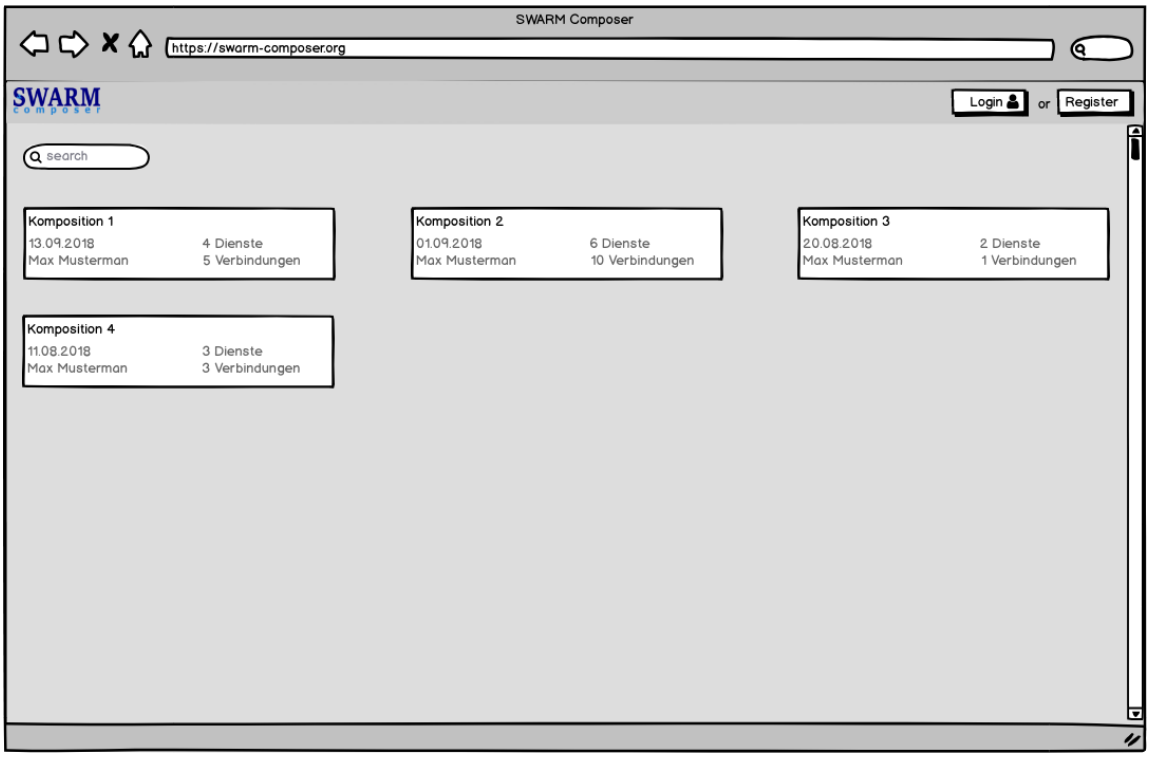


Abbildung 8.2: Dies ist die konzipierte Startseite. Nicht eingeloggte Nutzende bekommen alle öffentlich einsehbaren Kompositionen angezeigt. Durch das Drücken auf eine Komposition wird der Nutzende zur Bearbeitungsseite weitergeleitet.

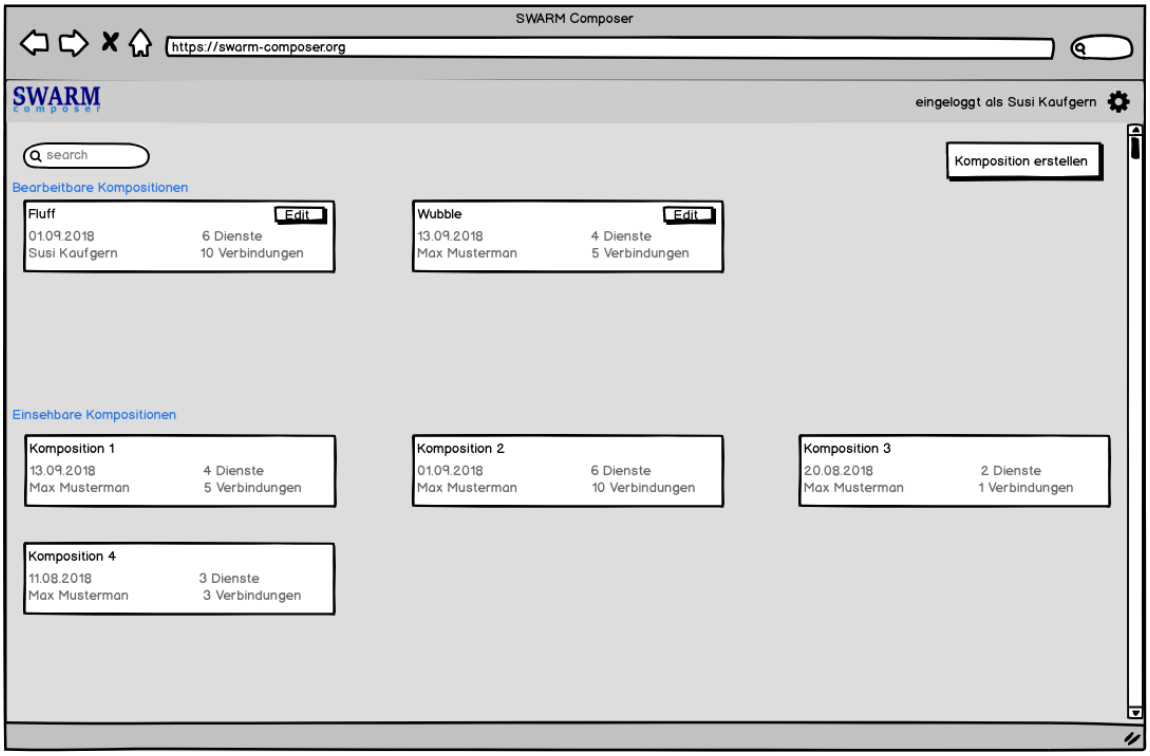


Abbildung 8.3: Eingeloggte Nutzende können zusätzlich zu den für sie sichtbaren Kompositionen auch noch für sie bearbeitbare Kompositionen getrennt einsehen.

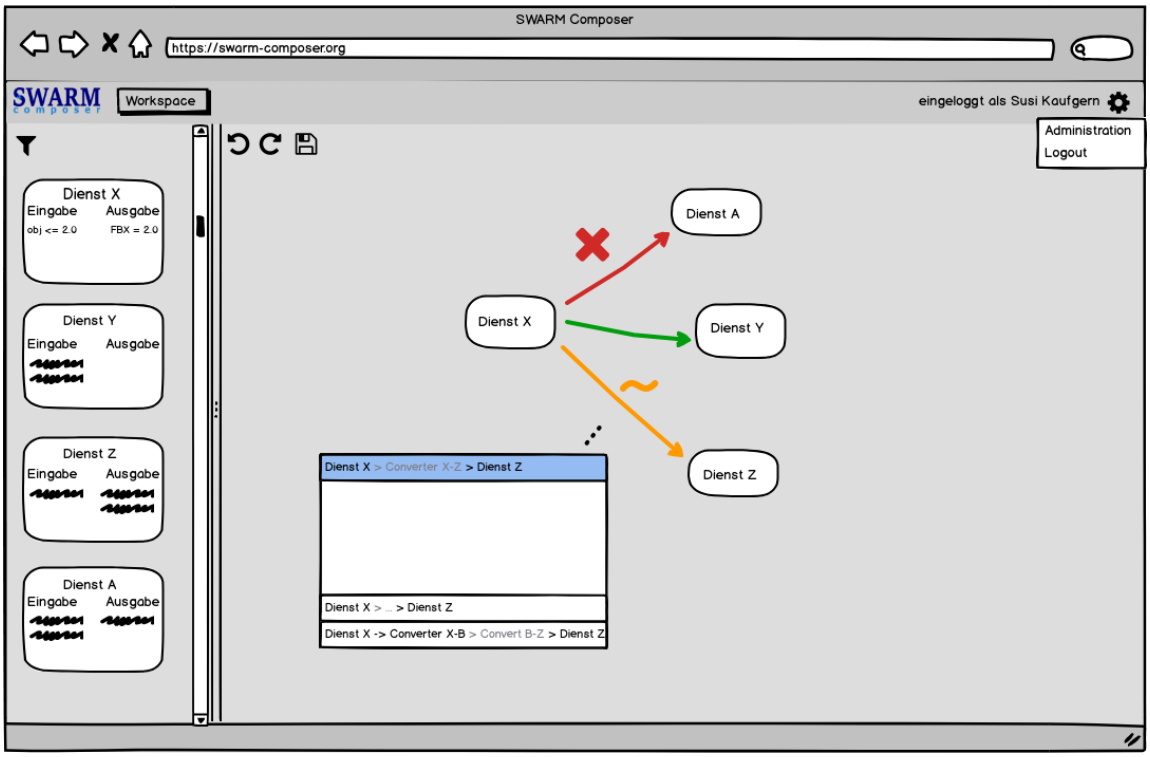


Abbildung 8.4: Hier können eingeloggte Nutzende eine zuvor gewählte Komposition bearbeiten. Dies beinhaltet das Einfügen und Löschen von Diensten durch Drag-and-Drop. Weiterhin werden sowohl inkompatible Verbindungen als auch Alternativvorschläge grafisch hervorgehoben.

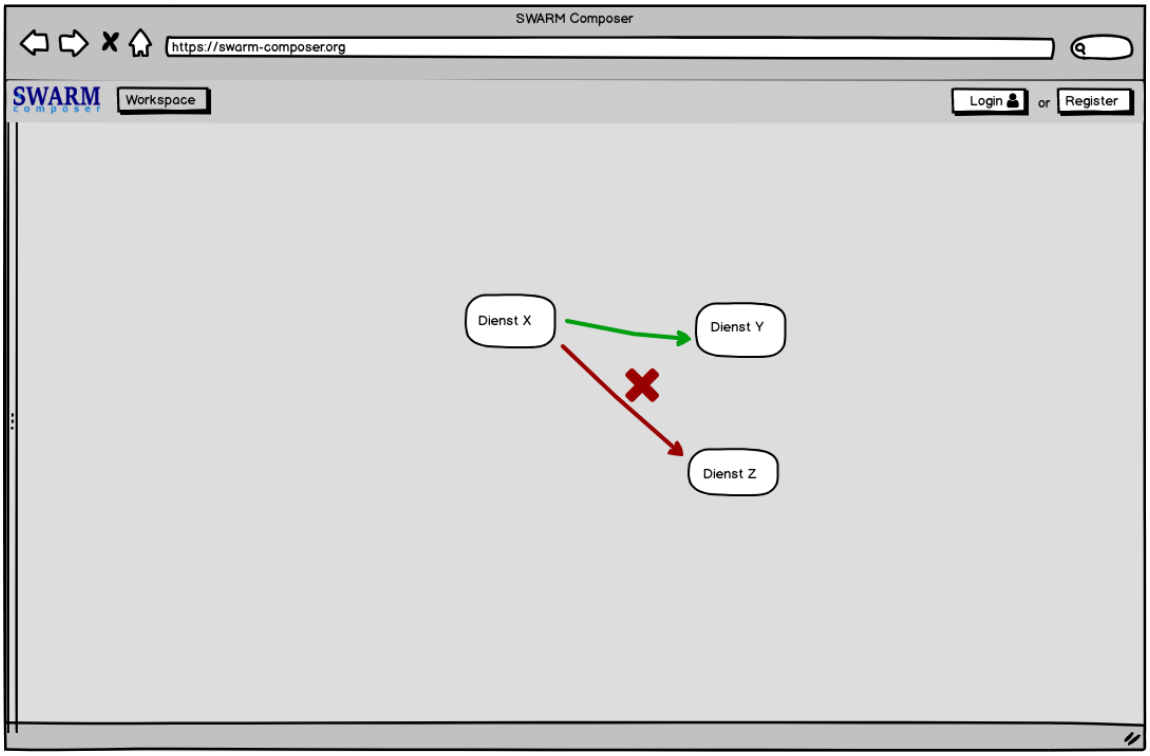


Abbildung 8.5: Nicht angemeldete Nutzende können Kompositionen anschauen, jedoch nicht bearbeiten.

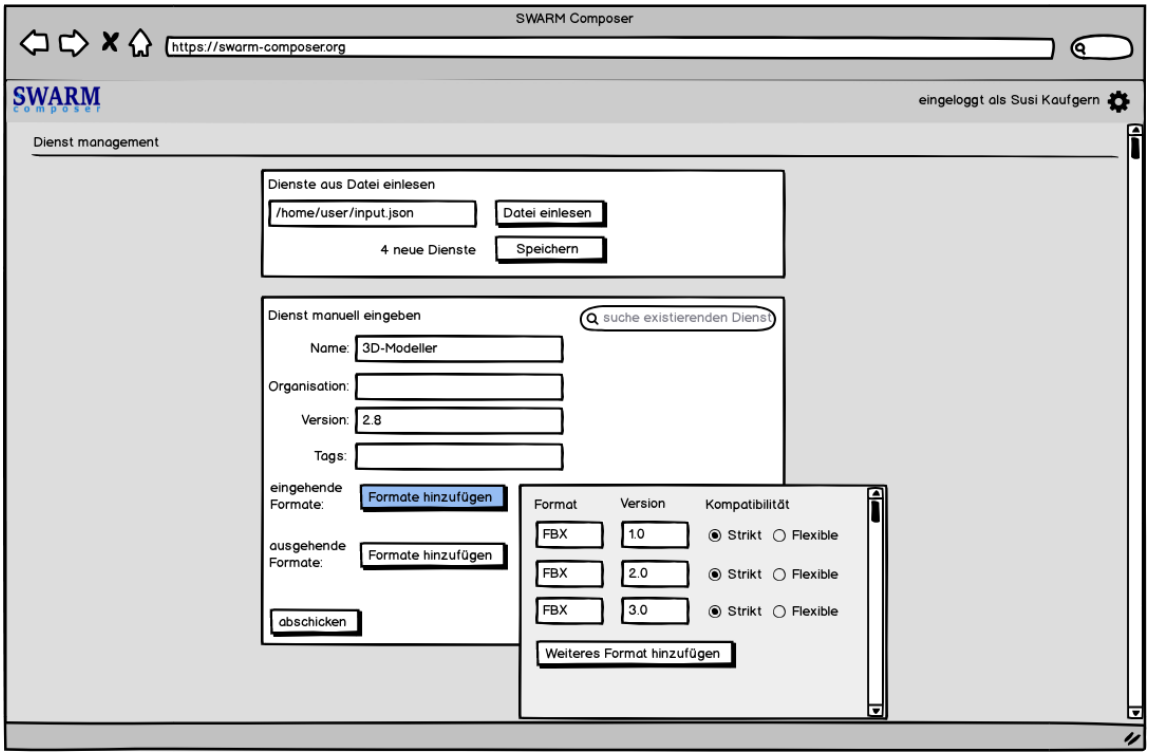


Abbildung 8.6: Hier kann ein Administrierender neue Dienste in das System einpflegen. Es besteht die Möglichkeit, mehrere Dienste aus einer JSON Datei einzulesen oder einen neuen manuell einzupflegen. Mit dem Suchfeld kann nach Diensten im System gesucht werden und über die gleiche Maske verändert werden.

Kapitel 9

Qualitätsanforderung

Die Tabelle 9.1 zeigt an, welche Eigenschaften der Software bei der Entwicklung mit welcher Priorität betrachtet werden.

	sehr wichtig	wichtig	weniger wichtig	unwichtig
Robustheit			x	
Zuverlässigkeit	x			
Wartbarkeit		x		
Erweiterbarkeit		x		
Benutzerfreundlichkeit	x			
Effizienz			x	
Anpassbarkeit			x	
Kompatibilität			x	
Sicherheit		x		

Tabelle 9.1: Qualitätsanforderungen