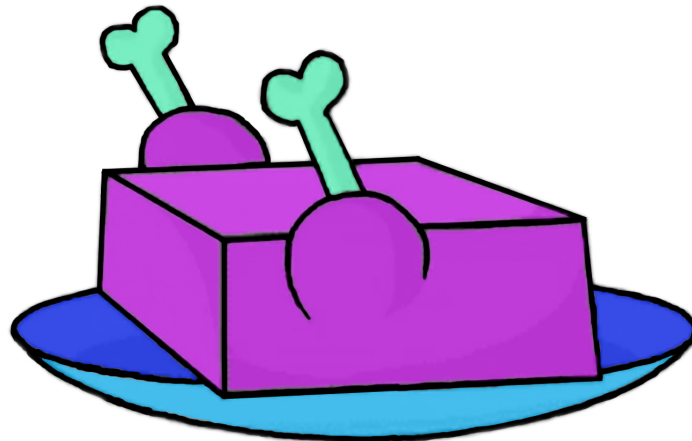


Projekt: kitovu
Anforderungsspezifikation



Florian Bruhin
florian.bruhin@hsr.ch

Méline Sieber
meline.sieber@hsr.ch

Nicolas Ganz
nicolas.ganz@hsr.ch

4. März 2018

Änderungsgeschichte

Datum	Version	Änderung	AutorIn
04.03.2018	1.0	Dokument erstellt, Grundgerüst von Template übernommen	Méline Sieber

1 Einführung

Dieses Dokument beschreibt, was *kitovu* genau ist. Danach veranschaulicht diese Anforderungsspezifikation, für wen der *kitovu*-Client gedacht ist. Das erfolgt anhand von "Use Cases", einerseits in einem kurzen Format (*brief*), andererseits in einer ausführlichen Beschreibung (*fully dressed*). Weitere Anforderungen detaillieren, welche Qualitätsmerkmale und Schnittstellen verwendet werden sowie geltende Randbedingungen.

1.1 Gültigkeitsbereich

Die vorliegende Anforderungsspezifikation ist für das Engineering Projekt im Frühlingssemester 2018 gültig. Falls dem Projekt grössere Veränderungen widerfahren, wird das Dokument dementsprechend angepasst. Umfassende Änderungen werden am Anfang des Dokuments protokolliert.

1.2 Referenzen

Die Anforderungsspezifikation ist eng verbunden mit der Domainanalyse und anderen Dokumenten. Die folgende Tabelle listet die wichtigsten Referenzen auf.

FIXME Verweise zum Projektplan einfügen, zur Domainanalyse u.a. Dokumenten, die wir mit der Anforderungsspezifikation abgeben

Confluence	https://jira.keltec.ch/wiki
Domainanalyse	FIXME
Draw.io	https://www.draw.io/
Github-Repository von <i>kitovu</i>	https://github.com/kitovu-bot/kitovu
JIRA	https://jira.keltec.ch/jira
Moodle	https://moodle.hsr.ch
OpenHSR Connect	https://github.com/openhsr/connect
Studentenportal	https://studentenportal.ch/
Switch AAI (Authentication and Authorization Infrastructure)	https://www.switch.ch/aai/

Beim Logo auf der Titelseite handelt es sich um eine stark überarbeitete Version eines GIFs (https://www.animateit.net/details.php?image_id=8990). Urheber und Copyright waren nicht auffindbar.

2 Allgemeine Beschreibung

2.1 Produktperspektive

2.2 Produktfunktion

Kitovu ist ein Client, der von verschiedenen Plattformen ausgewählte HSR-Unterrichtsmaterialien auf den eigenen Rechner synchronisiert. Er läuft auf allen gängigen Betriebssystemen und funktioniert nicht nur für den HSR-Skripteserver, sondern ist auch erweiterbar für verschiedene Plattformen.

Unser Projekt bindet primär den Skripteserver ein. Der Terminal-basierte Client funktioniert mittels Profilen zu unterschiedlichen Plattformen (Moodle, Skripteserver, Studentenportal). Pro Profil sind Verbindungsdaten und eventuelle Login-Credentials im Client hinterlegt. Die Daten-Synchronisation erfolgt immer nur von Server zu Client, erfolgreiche und misslungene Datentransfers werden protokolliert. Ein rudimentäres GUI dient als Proof-of-Concept.

Pro Profil lässt sich Folgendes definieren:

- welche Ordner/Dateien synchronisiert werden.
- welche Ordner/Dateien von der Synchronisation ausgeschlossen werden.
- wie mit Duplikaten/lokal bestehenden Dateien umgegangen wird.

Kitovu ist ausbaubar und damit modular: Zusätzlich zu den beiden Plattformen (Skripteserver; Moodle oder Studentenportal) können in zukünftigen Projekten beliebig viele Plattformen als separates Plugin bzw. Profil realisiert werden.

Optionale Features:

- Moodle und/oder das Studentenportal. Die Implementation von Moodle oder des Studentenportals ist abhängig von den weiter unten beschriebenen Risiken.
- Komplettes GUI, das der Funktionalität des Terminalprogramms entspricht.

2.3 Benutzercharakteristik

Studentinnen und Studenten verwenden *kitovu*, um ihre Unterrichtsmaterialien auf ihren Rechnern à jour zu halten. Sie verwenden verschiedene Betriebssysteme (Windows, macOS, Linux). Ihre Erfahrung mit der Kommandozeile ist unterschiedlich; manche verwenden sie nie, andere benutzen sie zur Standardinteraktion mit dem Betriebssystem.

Dozentinnen und Dozenten können ebenfalls *kitovu* verwenden, sie sind jedoch nicht die primäre Zielgruppe.

2.4 Einschränkungen

Da die Projekt-Zeitspanne kurz ist, ist die Kernfunktion von *kitovu* ein Client, der über die Kommandozeile bedient wird. Es ist vorerst nur eine rudimentäre grafische Benutzeroberfläche geplant. Falls jedoch genügend Zeit bleibt, baut das Team diese aus. Uns ist bewusst, dass wir damit einen Teil der Studierenden ausschließen, nicht alle können mit der Kommandozeile umgehen. Aufgrund der Zeitbeschränkung müssen wir das in Kauf nehmen, sehen es aber als erste Priorität, die wir implementieren, wenn bis zum Ende des Projekts genügend Zeit bleiben sollte.

Eine weitere Einschränkung sind das Studentenportal und Moodle. Die Gründe dazu beschreibt bereits der Projektplan ausführlich.

2.5 Annahmen

Eine wichtige Voraussetzung sind bereits bestehende Accounts. Wir gehen davon aus, dass die Studierenden bereits einen HSR-Account besitzen und sich damit sowohl per VPN von zu Hause als auch direkt an der HSR mit dem Skripteserver verbinden können. Das Studentenportal¹ verlangt einen separaten Account, der Zugriff auf Moodle² erfolgt via Switch-AAI, der Authentisierung- und Autorisierungsschnittstelle für alle Hochschulen³.

Wir müssen annehmen, dass ein Teil der Studentinnen und Studenten darin vertraut ist, die Kommandozeile zu bedienen. Die Konfiguration von *kitovu* erfolgt über ein einzelnes Konfigurationsfile - wir müssen ebenfalls davon ausgehen, dass die Studierenden damit umgehen können.

2.6 Abhängigkeiten

Kitovu steht und fällt mit der Anbindung an die Plattformen, also den HSR-Skripteserver sowie die optionalen Plattformen Moodle oder Studentenportal.

¹<https://studentenportal.ch/>

²<https://moodle.hsr.ch>

³<https://www.switch.ch/aai/>

3 Funktionale Anforderungen

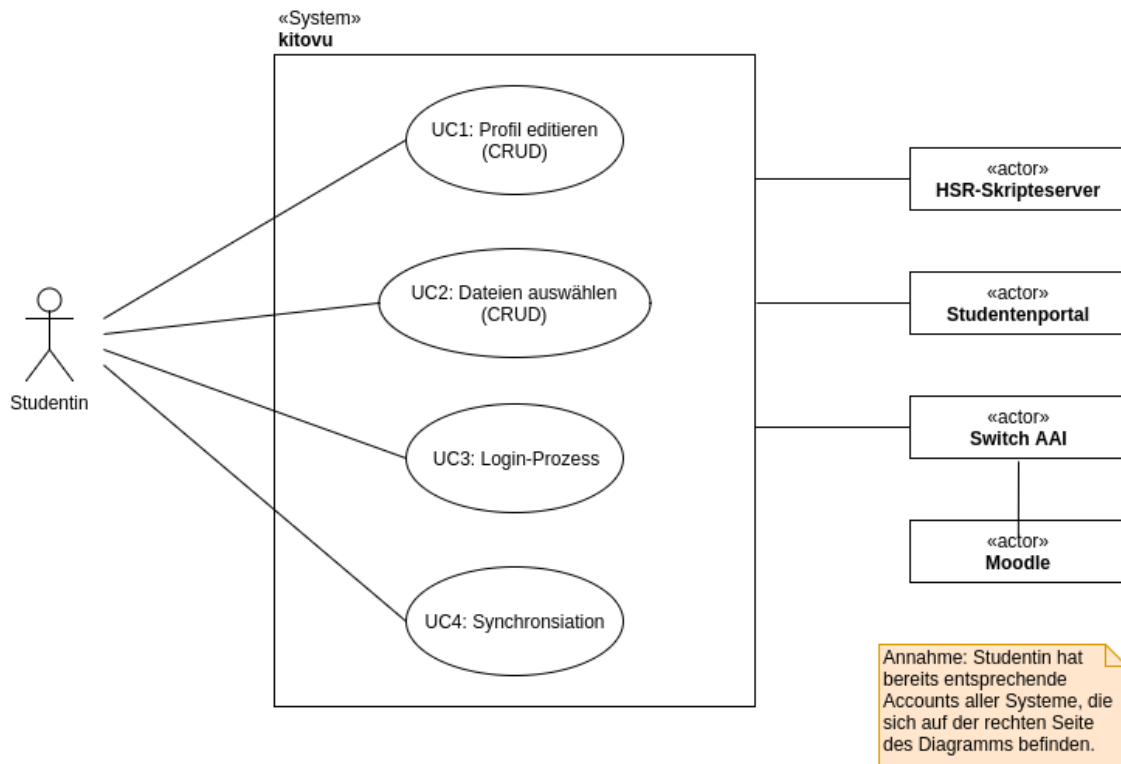
3.1 Akteure und Stakeholder

FIXME: wer sind die Stakeholder?

Aktor	Beschreibung
Student	HSR-Student, der über einen HSR-Account verfügt.
Skripteserver	HSR-Plattform, in der Dozierende Unterrichtsmaterialien zur Verfügung stellen.
Studentenportal	Portal für eigene Materialien der HSR-Studierenden.
Switch-AAI	Supporting Actor; ermöglicht Zugriff auf Moodle

3.2 Use-Case-Diagramm

FIXME insert definitive use case diagram, schön anordnen



3.3 Beschreibungen (Brief)

Use Case 1: Profil editieren (CRUD): Die Studentin legt in einer Konfigurationsdatei die Plattform und die entsprechenden Zugangsdaten fest, mit der auf die externe Plattform zugegriffen wird.

Use Case 2: Zu synchronisierende Dateien editieren (CRUD): Die Studentin konfiguriert, welche Ordner und Dateien synchronisiert werden sollen.

Use Case 3: Verbindung erstellen: Die Studentin verifiziert sich bei der externen Plattform.

Use Case 4: Synchronisation der Dateien: Die Studentin synchronisiert die Dateien von der externen Plattform auf ihren Rechner.

3.4 Beschreibungen (Fully Dressed)

FIXME Tabelle in schön

3.4.1 UC1: Profil editieren

Goal	Der Student editiert ein Profil zu einer externen Plattform: CRUD (Create, Read Update, Delete)
Level	FIXME
Primary Actor	Student
Trigger	Der Student möchte aktuelle Unterrichtsmaterialien auf seinem Rechner haben.
Stakeholders and Interests	FIXME
Preconditions	Der Student besitzt einen Account der externen Plattform mit entsprechenden Login-Daten.
Postconditions	FIXME
Main Success Scenario	Der Student erstellt ein Profil zu einer externen Plattform in <i>kitovu</i> . Er kann die Zugangsdaten erstellen, speichern, abändern oder das Profil ganz entfernen. Die eingegebenen Daten werden in <i>kitovu</i> gespeichert.
Extensions	Die externe Plattform verlangt, dass sich der Student pro Session authentisiert (Moodle).
Special Requirements	asdf
Frequency of Occurrence	asdf
Open Issues	Keine.

3.4.2 UC2: Dateien auswählen

Goal	Die Studentin wählt aus, welche Dateien sie auf ihren Rechner synchronisieren möchte (CRUD)-
Level	FIXME
Primary Actor	Studentin
Trigger	Die Studentin möchte nur bestimmte Dateien und Ordner auf ihrem Rechner haben.
Stakeholders and Interests	FIXME
Preconditions	Die Studentin besitzt einen Account der externen Plattform. Sie hat bereits ein Profil der externen Plattform in <i>kitovu</i> hinterlegt.
Postconditions	UC3 ist erfolgreich.
Main Success Scenario	Die Studentin wählt die zu synchronisierenden Daten aus, startet den Synchronisationsprozess und verfügt danach über die von ihr ausgewählten Unterrichtsmaterialien auf ihrem Rechner.
Extensions	Sie hat bereits eine
Special Requirements	asdf
Frequency of Occurrence	Zu Beginn jedes Semesters, wenn die neuen Module beginnen.
Open Issues	asdf

3.4.3 UC3: Verbindung erstellen

Goal	Um auf die Unterrichtsmaterialien zuzugreifen, muss der Student eine Verbindung zu externen Plattform aufnehmen.
Level	FIXME
Primary Actor	Der Student
Trigger	https://github.com/kitovu-bot/kitovu
Stakeholders and Interests	https://jira.keltec.ch/jira
Preconditions	Der Student hat in UC1 die Profildaten der externen Plattform hinterlegt.
Postconditions	keine.
Main Success Scenario	Der Student verbindet sich erfolgreich zur externen Plattform.
Extensions	https://www.switch.ch/aai/
Special Requirements	Der Student verfügt über einen Account der externen Plattform.
Frequency of Occurrence	FIXME
Open Issues	FIXME

3.4.4 UC4: Synchronisation der Dateien

Goal	Die Studentin synchronisiert alle benötigten Unterrichtsmaterialien auf ihren Computer.
Level	FIXME
Primary Actor	Die Studentin
Trigger	Die Studentin möchte aktuelle Unterrichtsmaterialien auf ihrem Computer lokal abspeichern.
Stakeholders and Interests	FIXME
Preconditions	Die Studentin hat erfolgreich eine Verbindung zur externen Plattform hergestellt.
Postconditions	FIXME
Main Success Scenario	FIXME
Extensions	FIXME
Special Requirements	FIXME
Frequency of Occurrence	Rund jede Woche, wenn die Dozenten erneut neue Unterrichtsmaterialien bereitstellen.
Open Issues	FIXME

4 Weitere Anforderungen

4.1 Qualitätsmerkmale

Gemäss ISO 9126

4.1.1 Funktionalität (functionality)

Angemessenheit (suitability) FIXME

Richtigkeit (accuracy) FIXME

Interoperabilität (interoperability) FIXME

Sicherheit (security) FIXME

Ordnungsmässigkeit (functionality compliance) FIXME

4.1.2 Zuverlässigkeit (reliability)

Reife (maturity) FIXME

Fehlertoleranz (fault tolerance) FIXME

Wiederherstellbarkeit (recoverability) FIXME

Konformität (reliability compliance) FIXME

4.1.3 Benutzbarkeit (usability)

Verständlichkeit (understandability) FIXME

Erlernbarkeit (learnability) FIXME

Bedienbarkeit (operability) FIXME

Attraktivität (attractiveness) FIXME

Konformität (usability compliance) FIXME

4.1.4 Effizienz (efficiency)

Zeitverhalten (time behaviour) FIXME

Verbrauchsverhalten (resource utilization) FIXME

Konformität (efficiency compliance) FIXME

4.1.5 Änderbarkeit (maintainability)

Analysierbarkeit (analyzability) FIXME

Modifizierbarkeit (changeability) FIXME

Stabilität (stability) FIXME

Testbarkeit (testability) FIXME

Konfromität (maintainability compliance) FIXME

4.1.6 Übertragbarkeit (portability)

Anpassbarkeit (adaptability) FIXME

Installierbarkeit (installability) FIXME

Koexistenz (co-existence) FIXME

Austauschbarkeit (replaceability) FIXME

Konformität (Portability compliance) FIXME

4.2 Schnittstellen

Folgende Schnittstellen werden im Projekt *kitovu* zu Verfügung gestellt oder verwendet:

Benutzeroberfläche Die Benutzeroberfläche ist die Schnittstelle, mit welcher die meisten Endbenutzer mit der Applikation interagieren wird. Diese bietet einen einfachen Weg, um die Daten zu synchronisieren.

Command-Line-Programm Alternativ bieten wir ein Command-Line-Programm, welches ermöglicht über einfache Befehle die Synchronisation zu starten. Dies ist besonders für Automatisierung oder für Endbenutzer, welche die Kommandozeile bevorzugen, gedacht.

Externe Schnittstellen Wir verwenden als externe Schnittstellen das Studentenportal, das Moodle und den Skripteserver, sowie das Switch AAI für die Authorisierung des Moodles.

4.3 Randbedingungen

Dies technischen Randbedingungen wurden für das Projekt *kitovu* festgelegt:

Python Version 3.6

Betriebssystem Hauptsächlich Linux, aber auch Windows und macOS unterstützt