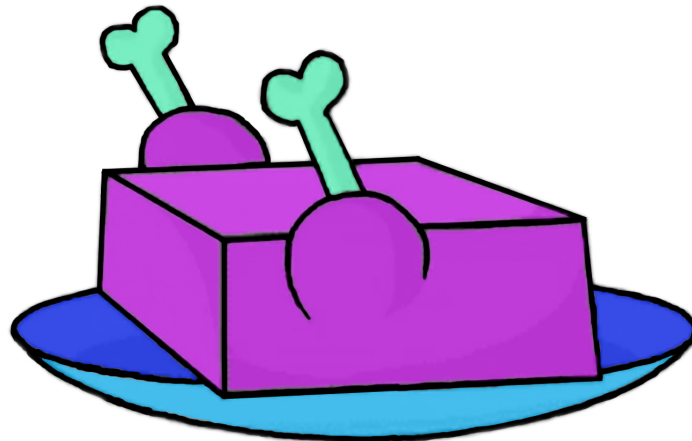


Projekt: kitovu
Anforderungsspezifikation



Florian Bruhin
florian.bruhin@hsr.ch

Méline Sieber
meline.sieber@hsr.ch

Nicolas Ganz
nicolas.ganz@hsr.ch

6. März 2018

Änderungsgeschichte

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

1 Einführung

Dieses Dokument beschreibt, was *kitovu* genau ist. Danach veranschaulicht diese Anforderungsspezifikation, wer *kitovu* verwendet. 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 mit der Domainanalyse und anderen Dokumenten verbunden. 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 Kommandozeilen-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.¹
- Komplettes GUI, das der Funktionalität des Kommandozeilen-Clients 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 ihrem 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. Falls genügend Zeit bleiben sollte, erweitern wir den grafischen Client auf die Funktionalität des Terminal-Clients.

¹Die Implementation von Moodle oder des Studentenportals ist abhängig von verschiedenen Risiken, die der Projektplan genauer ausführt.

Weitere Einschränkungen 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 Studentenportal² verlangt einen separaten Account, der Zugriff auf Moodle³ erfolgt via Switch-AAI, der Authentisierung- und Autorisierungsschnittstelle für alle Schweizer Hochschulen⁴.

Wir müssen annehmen, dass ein Teil der Studentinnen und Studenten 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. Folglich schliesst ein Kommandozeilen-basierter Client einen Teil der Studierenden aus.

2.6 Abhängigkeiten

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

²<https://studentenportal.ch/>

³<https://moodle.hsr.ch>

⁴<https://www.switch.ch/aai/>

3 Funktionale Anforderungen

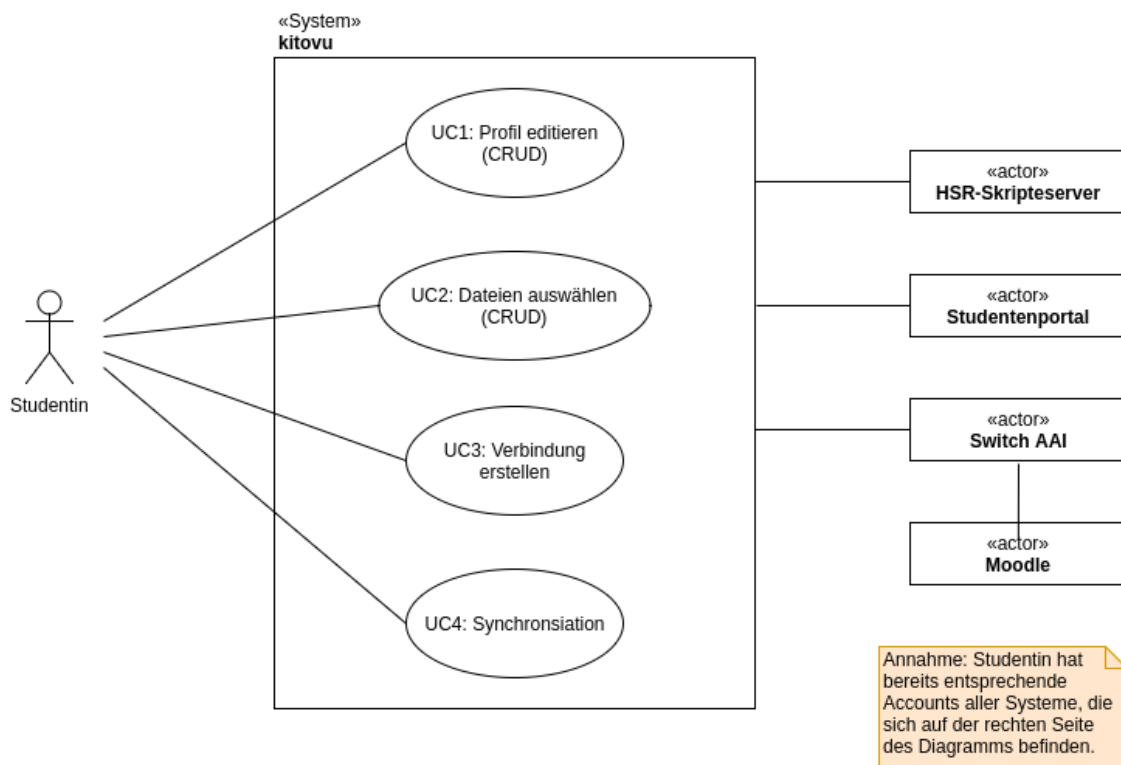
3.1 Aktoren und Stakeholder

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
Moodle	Indirekter Actor, Zugriff nur via Switch-AAI

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: Dateien auswählen (CRUD): Die Studentin konfiguriert, welche Ordner und Dateien synchronisiert werden sollen.

Use Case 3: Verbindung erstellen: Die Studentin stellt eine Verbindung zur externen Plattform her und authentisiert sich dort.

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

3.4 Beschreibungen (Fully Dressed)

FIXME Tabelle in schön- Weggelassen: Level, weil macht irgendwie keinen Sinn

3.4.1 UC1: Profil editieren

Goal	Der Student editiert das Profil zu einer externen Plattform: CRUD (Create, Read Update, Delete).
Primary Actor	Student
Trigger	Der Student möchte aktuelle Unterrichtsmaterialien auf seinem Rechner haben.
Stakeholders and Interests	Der Student: legt eine neue Plattform an. Das System: informiert den Studenten, welche Plattformen unterstützt werden und stellt sicher, dass die Plattform bereits implementiert ist.
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	Moodle: Die externe Plattform verlangt, dass sich der Student pro Session authentisiert. HSR-Skripteserver: Die externe Plattform verlangt, dass sich der Student im HSR-Netz befindet oder bereits eine VPN-Verbindung zur Plattform aufgenommen hat.
Frequency of Occurrence	Selten, nach Bedarf.
Open Issues	Keine.

3.4.2 UC2: Dateien auswählen

Goal	Die Studentin wählt aus, welche Dateien sie auf ihren Rechner synchronisieren möchte (CRUD). Sie kann ausschliessen, welche Dateien nicht synchronisiert werden.
Primary Actor	Studentin
Trigger	Die Studentin möchte nur bestimmte Dateien und Ordner auf ihrem Rechner haben.
Stakeholders and Interests	Die Studentin: weiss, welche Dateien sie synchronisieren möchte. Das System: kann die entsprechenden Dateien lokalisieren.
Preconditions	UC1 ist erfolgt: Die Studentin besitzt einen Account der externen Plattform. Sie hat bereits ein Profil der externen Plattform in <i>kitovu</i> hinterlegt. [FIXME linebreak] Sie weiss, welche Dateien sie synchronisieren möchte.
Postconditions	UC3 ist erfolgreich.
Main Success Scenario	Die Studentin definiert für das jeweilige Profil, welche Dateien sie synchronisiert haben möchte und welche ausgeschlossen werden.
Extensions	FIXME
Frequency of Occurrence	Zu Beginn jedes Semesters, wenn die neuen Module beginnen.

3.4.3 UC3: Verbindung erstellen

Goal	Um auf die Unterrichtsmaterialien zuzugreifen, muss der Student eine Verbindung zur externen Plattform herstellen.
Primary Actor	Der Student
Trigger	Der Student möchte auf eine externe Plattform zugreifen.
Stakeholders and Interests	Der Student: weiss, von welcher Plattform er welche Dateien möchte. Das System: kann die Verbindung herstellen.
Preconditions	Der Student hat in UC1 die Profildaten der externen Plattform hinterlegt.
Postconditions	keine.
Main Success Scenario	Der Student verbindet sich zur externen Plattform, die Authentisierung ist erfolgreich.
Extensions	Die Verbindung misslingt: [FIXME Bulletpoint] Nicht im HSR-Netz (HSR-Skripteserver): Die Verbindung misslingt, da sich der Student nicht im HSR-Netz befindet oder nicht per VPN verbunden ist. [FIXME Bulletpoint] Moodle-Probleme: der Zugriff auf Moodle misslingt aufgrund Plattform-eigener Probleme.
Special Requirements	Der Student verfügt über einen Account der externen Plattform.
Frequency of Occurrence	Rund jede Woche, wenn die Dozenten erneut neue Unterrichtsmaterialien bereitstellen.

3.4.4 UC4: Synchronisation der Dateien

Goal	Die Studentin synchronisiert alle benötigten Unterrichtsmaterialien auf ihren Computer.
Primary Actor	Die Studentin
Trigger	Die Studentin möchte aktuelle Unterrichtsmaterialien auf ihrem Computer lokal abspeichern.
Stakeholders and Interests	Die Studentin: möchte stets die Version der Dateien auf ihrem Rechner haben, die sie tatsächlich will. Das System: stellt sicher, dass die gespeicherten Dateien konsistent und aktuell sind. Es modifiziert keine Dateien, ohne dass die Studentin nicht davon wüsste.
Preconditions	UC3: Die Studentin hat erfolgreich eine Verbindung zur externen Plattform hergestellt.
Main Success Scenario	Die Studentin startet den Synchronisationsprozess. Danach verfügt sie über die von ihr ausgewählten Unterrichtsmaterialien auf ihrem Rechner. Eine Log-Datei protokolliert den Vorgang.
Extensions	Dateien schon vorhanden: Die Studentin legt in ihrem Profil fest, wie mit bereits existierenden Dateien umgegangen werden soll. Dabei kann sie wählen, ob lokal bestehende Dateien gleichen Namens umbenannt, durch eine neuere Server-Version ersetzt oder nichts synchronisiert werden soll. Sie kann auch entscheiden, bei jedem Konflikt von <i>kitovu</i> gefragt zu werden, wie sie für jede Datei entscheiden möchte. Synchronisation schlägt fehl: Die Synchronisation wird nicht durchgeführt, die Studentin erhält eine Benachrichtigung.
Frequency of Occurrence	Rund jede Woche, wenn die Dozenten erneut neue Unterrichtsmaterialien bereitstellen.

4 Weitere Anforderungen

4.1 Qualitätsmerkmale

Für die Qualitätsmerkmale haben wir uns für den Standard ISO/IEC 9126-1⁵ entschieden. Obwohl dieser bereits mit dem Standard ISO/IEC 25010⁶ überarbeitet wurde, verwenden wir die ältere Version, da wir damit bereits Erfahrung gesammelt haben.

4.1.1 Funktionalität (functionality)

Angemessenheit (suitability)

Die Funktionen von *kitovu* konzentrieren sich nur auf die Synchronisation der Dateien. Dadurch ist die Applikation sehr spezialisiert, aber trotzdem erweiterbar.

Richtigkeit (accuracy)

Es werden sämtliche Dateien, welche lokal nicht existieren, synchronisiert.

Bei der Erkennung von geänderten Dateien sind wir von den Daten abhängig, welche die externen Schnittstellen bieten, abhängig. Dies bedeutet, dass wenn zum Beispiel nur Änderungszeitpunkte eingesehen werden können, wir diese mit der entsprechenden Genauigkeit (je nach Datumsformat). Wenn nun etwas in der selben Sekunde ändert, erkennen wir dies nicht als Änderung. Zusätzlich können Änderungszeitpunkte verändert werden, was wir auch nicht verhindern, respektive umgehen, können.

Interoperabilität (interoperability)

Die externen Schnittstellen werden über abgekoppelte Module angesprochen. Dadurch sind diese einfach testbar und austauschbar, falls eine Schnittstelle nicht mehr benötigt wird oder eine neue gewünscht ist.

Sicherheit (security)

Bei *kitovu* werden Anmeldeinformationen (wie Benutzernamen, Passwörter und Login-Tokens) verschlüsselt auf dem System abgelegt.

Die Ablage der Daten und Konfigurationen geschieht nicht verschlüsselt, sprich es liegt in der Hand des Benutzers vorsichtig damit umzugehen. Falls Daten, welche vertraulich sind, synchronisiert werden muss der Benutzer sicherstellen, dass er es nicht z.B. auf einen öffentlichen Share synchronisiert.

Ordnungsmässigkeit (regularity)

Für die Ordnungsmässigkeit halten wir uns an das Bundesgesetz über den Datenschutz (DSG⁷). Wir speichern die Daten immer nur auf dem Computer des Benutzers. Die einzigen Daten, welche an dritte geht sind die Anmeldedaten für die entsprechende Schnittstelle, sowie die Namen der Dateien, welche synchronisiert werden müssen.

Sämtliche Änderungen an Daten werden von dem Benutzer ausgelöst und durch benennung der Funktionen wird verständlich gemacht, was geschieht.

Es wird nicht mit besonders schützenswerten Personendaten hantiert.

FIXME weiteres? evt. Richtlinien von Libraries oder Schnittstellen?

Konformität (functionality compliance)

Um die funktionale Konformität zu gewährleisten halten wir uns an die Konventionen in PEP 8⁸ und prüfen unseren Code mit Style-Checkern.

⁵ISO/IEC 9126-1 <https://www.iso.org/standard/22749.html>

⁶ISO/IEC 25010 <https://www.iso.org/standard/35733.html>

⁷<https://www.admin.ch/opc/de/classified-compilation/19920153/index.html>

⁸<https://www.python.org/dev/peps/pep-0008/>

4.1.2 Zuverlässigkeit (reliability)

Reife (maturity)

In 99% der Benutzer-Interaktionen sollte kein Fehler auftreten. Falls doch wird dieser abgefangen und eine verständliche und hilfreiche Fehlermeldung sollte dargestellt werden.

Davon ausgenommen sind Fehler bei der Synchronisation, welche von den externen Schnittstellen ausgelöst sind.

Fehlertoleranz (fault tolerance)

Falls ein Modul Fehlerhaft ausgeführt wird (zum Beispiel falls ein Server nicht erreichbar ist) wird dies dem Benutzer dargestellt, aber sämtlichen anderen Module werden ausgeführt.

Fehlerhafte Eingaben in der Konfiguration werden dem Benutzer verständlich kommuniziert.

Wiederherstellbarkeit (recoverability)

Falls eine Synchronisation nur zum Teil ausgeführt werden konnte, kann die Synchronisation erneut ausgeführt werden und die nur die fehlenden Teile werden versucht zu synchronisieren. Dadurch muss ein lange dauernde Synchronisierung nicht komplett erneut durchgeführt werden.

Falls sämtliche Dateien verloren gegangen sind muss nur das Profil erneut geschrieben werden und der Rest wird automatisch synchronisiert.

Konformität (reliability compliance)

FIXME Grad, in dem die Software Normen oder Vereinbarungen zur Zuverlässigkeit erfüllt

4.1.3 Benutzbarkeit (usability)

Verständlichkeit (understandability)

Da *kitovu* einen begrenzten, übersichtlichen Funktionsumfang hat werden wir die einzelnen Optionen und Aktionen in der Benutzeroberfläche, respektive der help-Option des Kommandozeilenprogramm, beschrieben.

Erlernbarkeit (learnability)

Die einzelnen Funktionen sind übersichtlich gruppiert und beschrieben.

Bedienbarkeit (operability)

Das erstellen des Profiles benötigt etwas aufwand, aber das eigentliche synchronisieren wird mit minimalem Aufwand ermöglicht.

Attraktivität (attractiveness)

Die Oberfläche und das Kommandozeilenprogramm sind übersichtlich strukturiert.

Konformität (usability compliance)

FIXME Grad, in dem die Software Normen oder Vereinbarungen zur Benutzbarkeit erfüllt.

4.1.4 Effizienz (efficiency)

Zeitverhalten (time behaviour)

Das Zeitverhalten ist abhängig, von den externen Schnittstellen und der Internet-Leitung.

Falls jedoch nichts geändert hat wird pro Modul immer nur ein Request abgesetzt, um die Änderung der Dateien bei der externen Schnittstelle herauszufinden. Je nach dem bietet die externe Schnittstelle eine solche Funktion nicht, wodurch wir für jede Datei ein Request absetzen müssen.

Verbrauchsverhalten (resource utilization)

Die CPU-Zeit ist minim gehalten, da die meiste Zeit in den Internet-Requests verbraucht wird.

Im RAM und am ende auf der Festplatte werden die Änderungszeitpunkte der einzelnen Dateien gehalten, was bei grossen Ordnern zu einigem Verbrauch führt. Zeitpunkte der einzelnen Dateien gehalten, was bei grossen Ordnern zu einigem Verbrauch führt.

Konformität (efficiency compliance)

FIXME Grad, in dem die Software Normen oder Vereinbarungen zur Effizienz erfüllt.

4.1.5 Änderbarkeit (maintainability)

Analysierbarkeit (analyzability)

Mit einem Debug Modus können zusätzlich die Rückgabewerte der Module gelogged werden. Dies ermöglicht eine Einschränkung des Problems auf entweder ein spezifisches Modul oder den Core-Teil selber.

Modifizierbarkeit (changeability)

Da die ganze Applikation modular aufgebaut ist, lässt diese sich leicht um neue Module erweitern oder bestimmte entfernen.

Stabilität (stability)

Eine grosse Testabdeckung (inklusive kompletter Abläufe) bieten eine Grundstabilität, so dass Änderungen bei bestimmten Funktionalitäten die Tests sofort reklamieren.

Testbarkeit (testability)

Dank dem modularen Aufbau können wir alle Module und den Core-Teil separat Testen. Dies ergänzen wir mit Integrations-Tests, welche den kompletten Ablauf testet.

Konformität (maintainability compliance)

FIXME Grad, in dem die Software Normen oder Vereinbarungen zur Änderbarkeit erfüllt.

4.1.6 Übertragbarkeit (portability)

Anpassbarkeit (adaptability)

Über neue Module lässt sich die Applikation einfach mit neuen Schnittstellen erweitern.

Installierbarkeit (installability)

Die Applikation lässt sich über ein pip-Paket ⁹ installieren. Dies installiert sämtliche Dependencies mit.

Koexistenz (co-existence)

Falls eine weitere Applikation an den selben Ort Dateien schreibt werden, je nach Konfiguration des Profils, die Dateien der anderen Applikation bei der nächsten Synchronisation einfach überschrieben. Je nach dem wie die andere Applikation die Dateien bearbeitet könnte es zu einer erneuten Synchronisation der Dateien führen, aber es funktioniert weiterhin.

Austauschbarkeit (replaceability)

Es kann nicht direkt eine andere Applikation ablösen, da wir einen Stand der Schnittstellen-Daten zwischenspeichern. Daher müssen die Daten zu Beginn frisch synchronisiert werden.

Konformität (Portability compliance)

FIXME Grad, in dem die Software Normen oder Vereinbarungen zur Übertragbarkeit erfüllt.

4.2 Schnittstellen

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

⁹<https://pypi.python.org/pypi/pip>

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