Berufsakademie Sachsen
Staatliche Studienakademie Leipzig

Pflichtenheft QuixSync

Projektarbeit
für Softwareprojekt im fünften Semester
in der Studienrichtung Informatik

Eingereicht von: Quentin Weber, 5000924
Philipp Ludwig, 5000964
Chris Sembritzki, 5000993
Seminargruppe: CS16-1

Inhaltsverzeichnis

1 Auftrag

1.1 Beteiligte Personen

Auftragnehmer:

Projektleiter:

Quentin Weber

- Quentin.Weber@cs16-1.ba-leipzig.de

Projektteam:

Philipp Ludwig

- Philipp.Ludwig@cs16-1.ba-leipzig.de

Chris Sembritzki

- Chris-Thomas.Sembritzki@cs16-1.ba-leipzig.de

Auftraggeber:

Christian Heller

- Christian.Heller@ba-leipzig.de

1.2 Auftragsbeschreibung

Das Ziel des Projektes ist das Erstellen eines Tools zur Synchronisation von Verzeichnissen und Dateien. Dies soll lokal auf einem Rechner oder über das Netzwerk erfolgen. Dem Nutzer wird für diesen Zweck eine grafische Oberfläche zur Verfügung gestellt.

1.3 Meilensteine

Lastenheft/Thema:	Freitag, den 05.10.2018 um 12:00 Uhr
Pflichtenheft:	Freitag, den 19.10.2018 um 12:00 Uhr
Softwareprototyp:	Freitag, den 07.12.2018 um 12:00 Uhr
Präsentation:	Freitag, den 14.12.2018 um 08:00 Uhr

1.4 Dokumentation

Die Dokumentation und Versionierung erfolgt über GitHub (https://github.com/BuckUbel/QuixSync).

2 Anforderungen

Die Anforderungen werden in Form von User Stories dargestellt. Diese beschreiben einen spezifischen Anwendungsfall, welcher von der Anwendung ausgelöst werden soll.

2.1 Pflichtanforderungen

Als Anwender möchte ich einen konkreten Ordner mit einem anderen vergleichen. Als Ergebnis dieses Vergleiches sollen mir die Unterschiede aufgelistet werden.

Als Anwender möchte ich einen bestimmten Ordner an verschiedenen Stellen innerhalb des Dateisystems ablegen. Diese verschiedenen Ordner sollen denselben Informationsgehalt und Struktur besitzen.

Als Anwender möchte ich eine graphische Benutzeroberfläche für die Konfiguration und Bedienung des Programmes verwenden.

2.2 Wunschanforderungen

Als Anwender möchte ich nicht nur lokale, also auf dem aktuellen Rechner erreichbare, Ordner synchronisieren. Ein Abgleich zweier Ordner über FTP wäre wünschenswert.

Als Anwender brauche ich einen automatisierten Abgleich der zu synchronisierenden Ordner. Das Zeitintervall dafür kann selbst gewählt werden. Dies sorgt dafür, dass ein manuell angestoßener Abgleich nicht mehr zwingend durchgeführt werden muss, um die beiden Ordner auf dem aktuellen Stand zu halten.

2.3 künftige Anforderungen

Die nachfolgenden User Stories beschreiben Anforderungen, die in der Zukunft der Anwendung hinzugefügt werden könnten.

Als Anwender möchte ich eine auf UNIX-Systemen optimierte Dateisynchronisation durchführen.

2.4 Use-Case-Diagramme

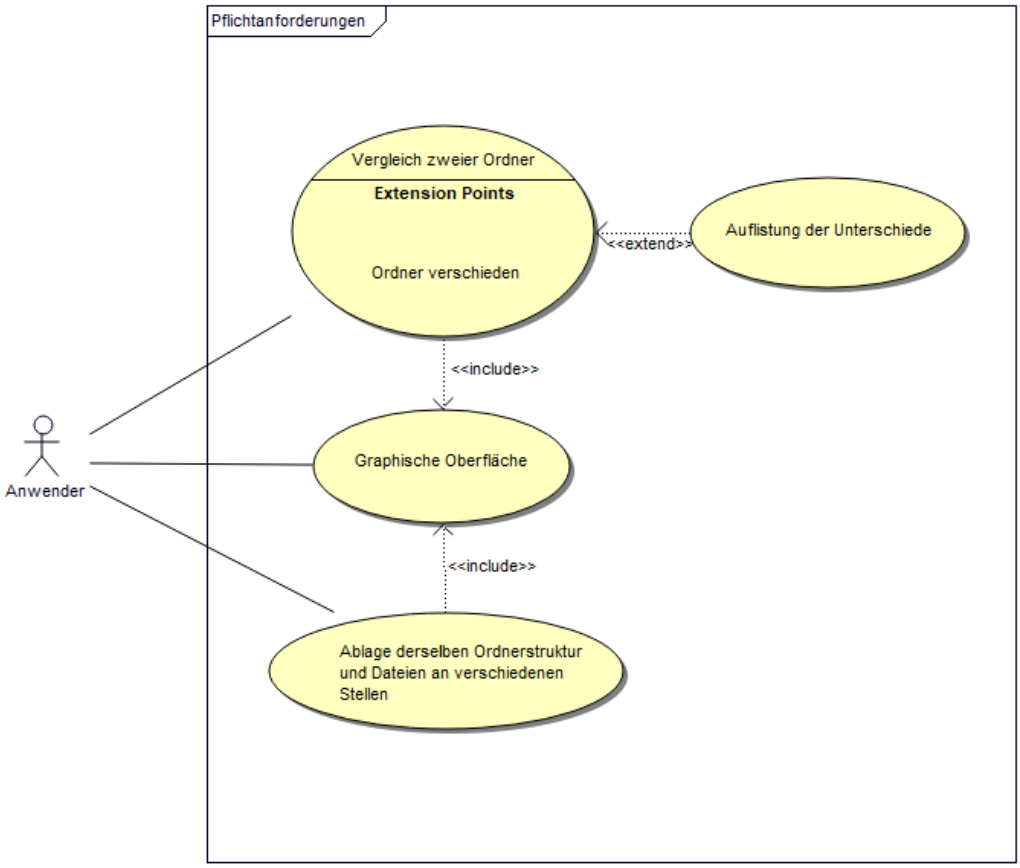

Abbildung 1 Pflichtanforderungen

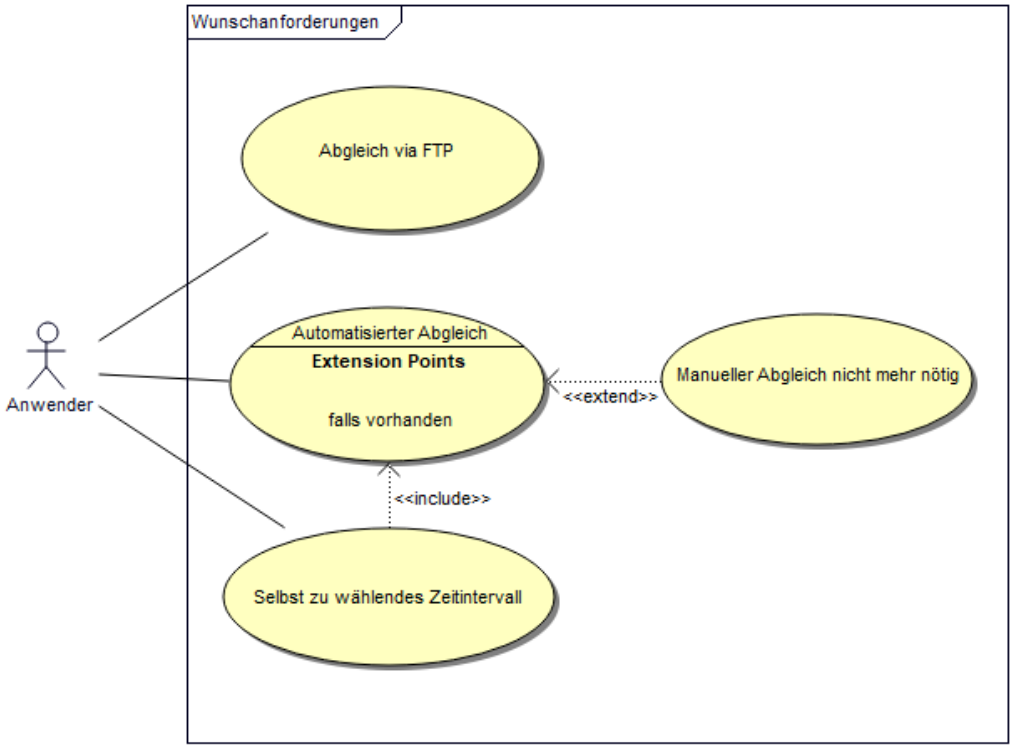

Abbildung 2 Wunschanforderungen

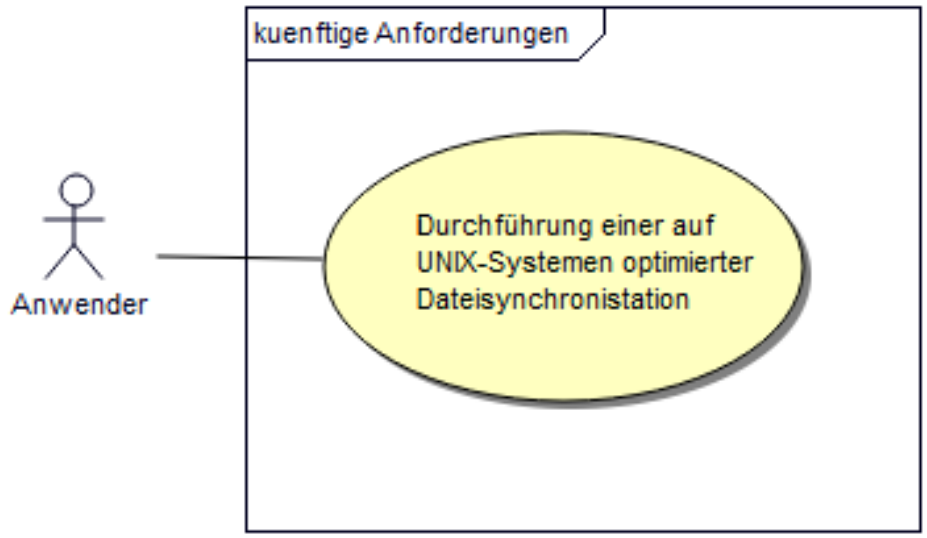

Abbildung 3 künftige Anforderungen

3 Produkteinsatz

3.1 Anwendungsbereiche

Dieses Programm hat die Funktion zwei Verzeichnisse zu analysieren, zu vergleichen und schlussendlich zu synchronisieren. Die Synchronisation eines Ordners kann auch als Backup für eben jenes Verzeichnis dienen. Dies kann über im Dateisystem eingebundene Ordner erfolgen.

3.2 Zielgruppen

Dieses Programm ist besonders für Personen interessant, die häufig auf unterschiedlichen Rechnern arbeiten müssen und den aktuellen Stand eines Ordners auf allen Geräten zur Verfügung gestellt haben wollen.

3.3 Betriebsbedingung

Der Benutzer muss ein Desktop-Betriebssystem mit der aktuellen Version von Java 8 verwenden. Allerdings wird im Rahmen dieses Projektes die Synchronisierung ausschließlich auf Windows 10 getestet.

Soll eine FTP-Verbindung erstellt werden, muss eine Netzwerkverbindung zu dem entsprechenden FTP-Server realisiert werden können.

4 Produktübersicht

4.1 Kurze Produktübersicht

QuixSync wird über eine grafische Oberfläche bedient. Diese soll dem Nutzer folgende Funktionen zur Verfügung stellen:

- Auswählen zweier Verzeichnisse (Quellverzeichnis, Zielverzeichnis)
- Erstellen einer Indexdatei für ein Verzeichnis
- Vereinfachtes Präsentieren einer Indexdatei
- Erstellen einer Vergleichsdatei zwischen zwei Indexdateien
- Vereinfachtes Präsentieren einer Vergleichsdatei
- Starten einer weichen Synchronisation aufgrund einer Vergleichsdatei
- Starten einer harten Synchronisation aufgrund einer Vergleichsdatei
- Starten einer kompletten Synchronisation
  (Erstellen und Vergleich von Indexdateien mit inbegriffen)
- Setzen von Nutzer-spezifischen Einstellungen

4.2 UML-Diagramme

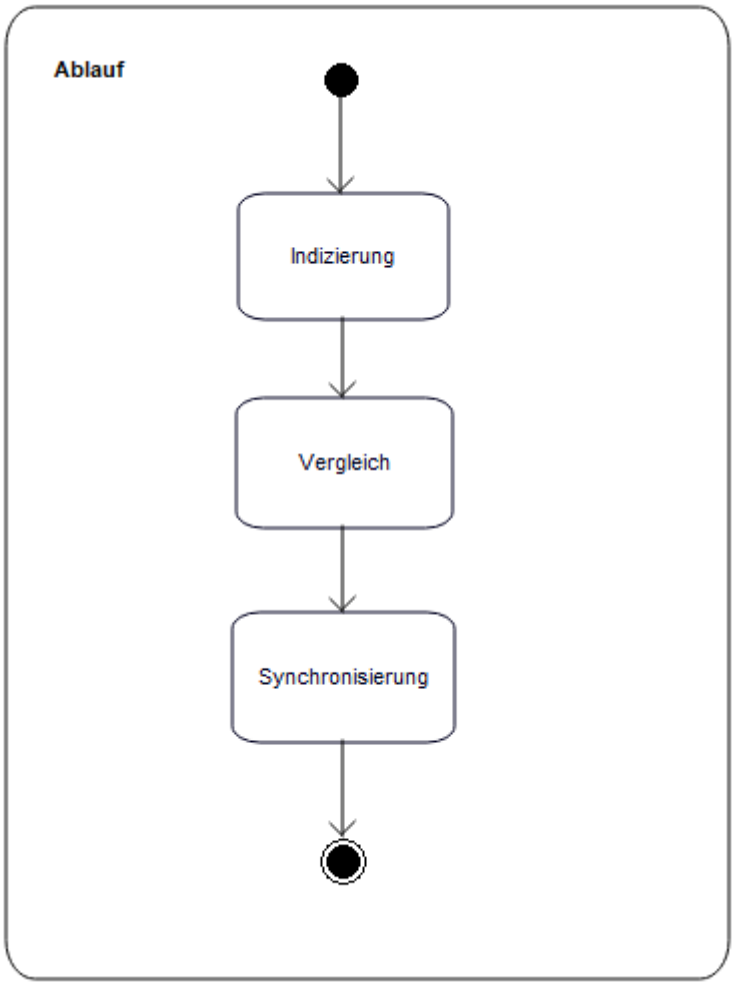

Abbildung 4 Ablauf einer kompletten Synchronisation

Parameter: Verzeichnis

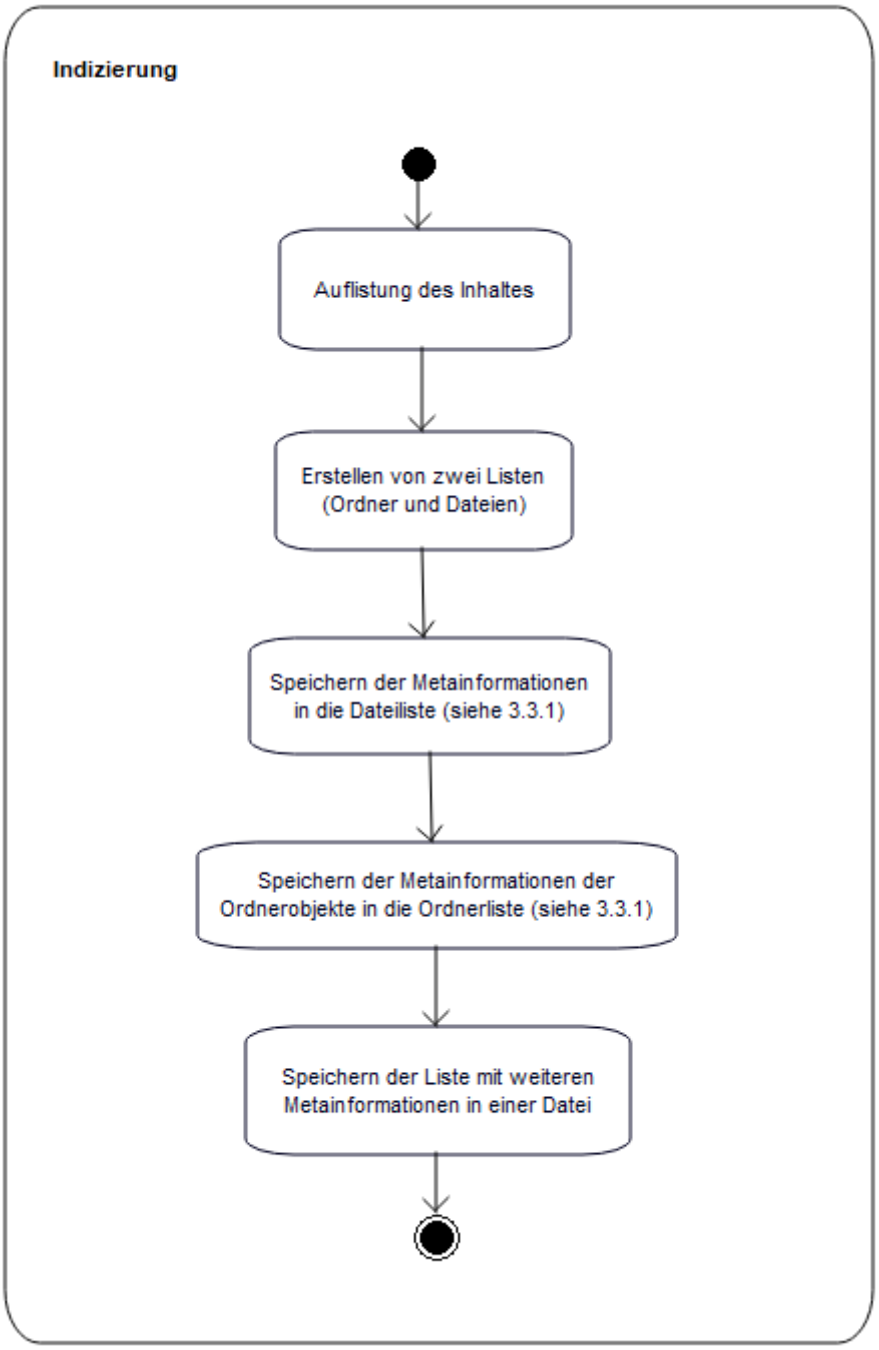

Abbildung 5 Indexierung

Parameter: zwei Indizierungsdateien, isHardSync

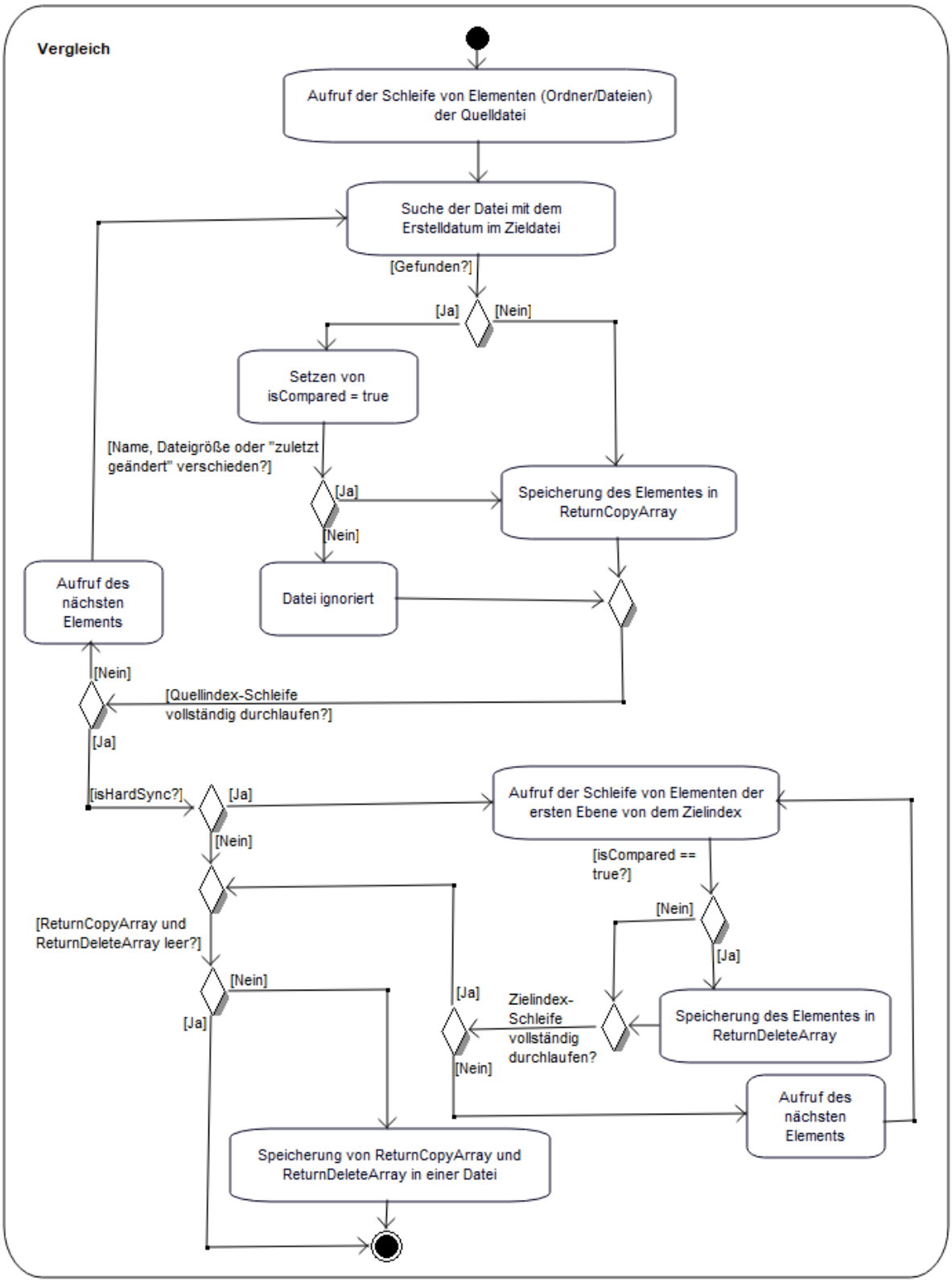

Abbildung 6 Vergleich

Parameter: Vergleichsdatei

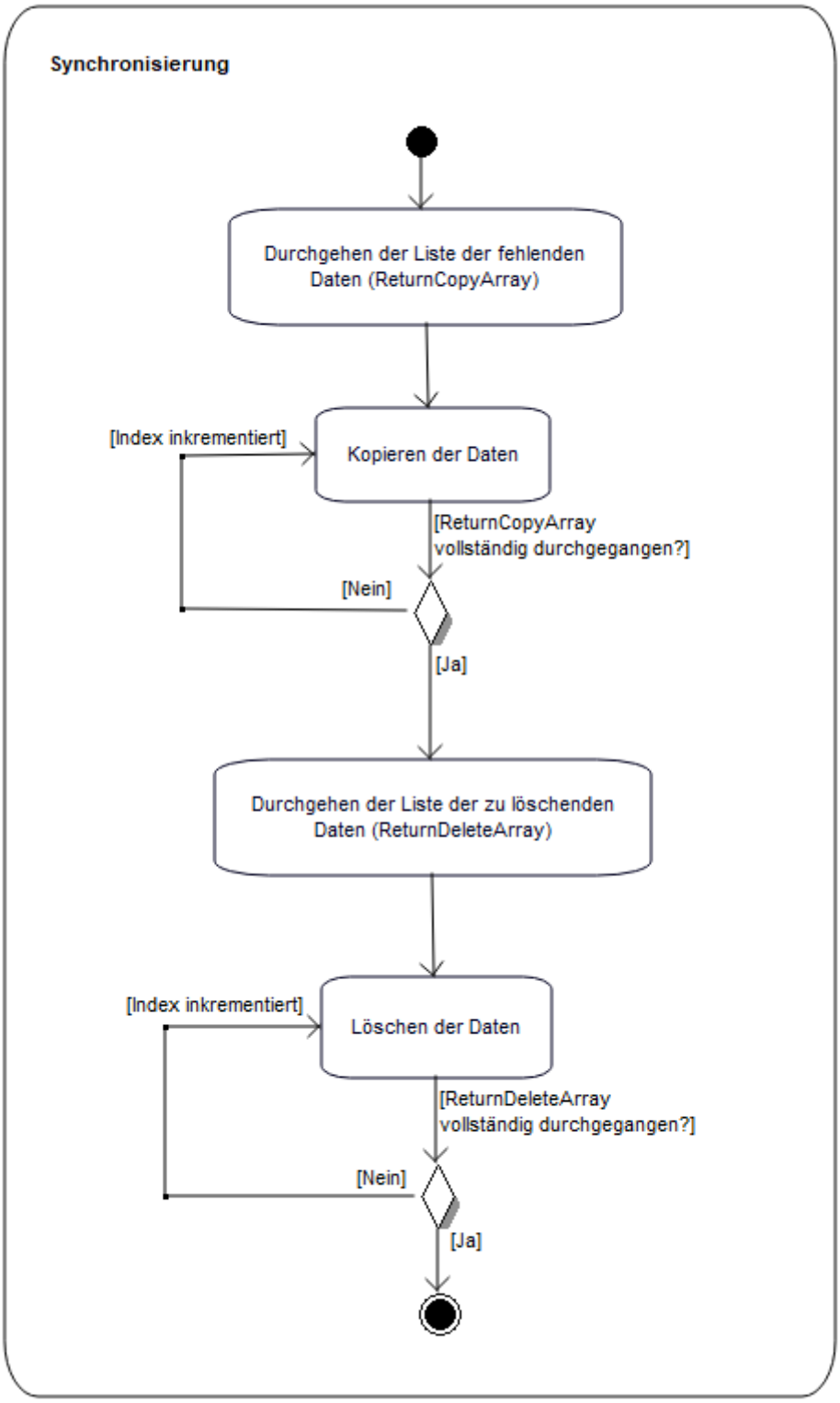

Abbildung 7 Synchronisierung

4.3 ER Diagramme

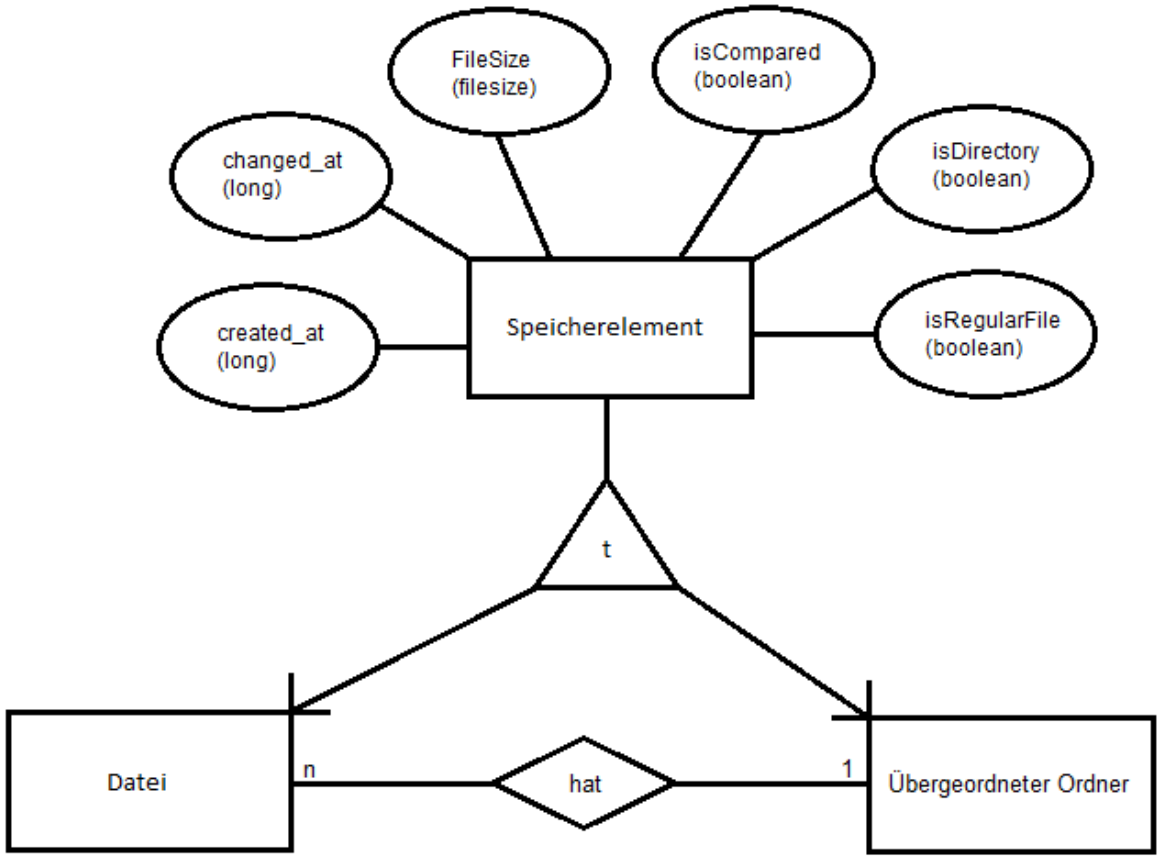

Abbildung 8 Speicherelement (StorageElement)

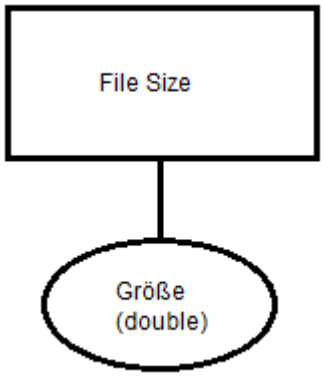

Abbildung 9 FileSize

5 Produktfunktionen

5.1 Analysieren

In diesem Punkt wird ein Verzeichnis indiziert. Das bedeutet man gibt ein Verzeichnis an, zu welchem eine Index-Datei erzeugt wird. Diese beinhaltet die Information zu jeder vorhandenen Datei und jedes Unterverzeichnisses. Zusätzlich werden auch Metadaten, wie das Erstelldatum, das Änderungsdatum und die Dateigröße gesammelt.

Zur Erstellung der Indexdatei wird, mittels einer rekursiven Funktion, das angegebene Verzeichnis durchsucht. Die beinhalteten Elemente werden zunächst innerhalb des Programms in zwei Listen gespeichert. Eine enthält die Dateien und eine die Verzeichnisse. Somit erfolgt eine Trennung dieser beiden Datentypen. Die beiden Listen werden nach der Befüllung in die Indexdatei geschrieben, wie auch die Anzahl der Dateien und der Speichergröße.

5.2 Vergleichen

Für den Vergleich benötigt man nun zwei dieser Indexdateien. Diese enthalten die Informationen über das Ziel- und das Quellverzeichnis. Die Quell-Indexdatei wird sequenziell, also Element für Element, abgearbeitet. Dabei wird überprüft, ob dieses Element auch in der Ziel-Indexdatei zu finden ist.
Als Grundlage für den Vergleich dient das Erstelldatum. Während sich der Name jederzeit durch Umbenennung ändern kann, bleibt die Erstellzeit immer gleich. Wenn ein Element gefunden wurde, welches im Ziel fehlt oder verändert wurde, wird dessen absoluter Pfad zum angegebenen Verzeichnis in einer neuen Liste (ReturnCopyArray) gespeichert. Diese wird später genutzt um diese Elemente in das Zielverzeichnis zu kopieren. Weiterhin bekommt das Element in der Ziel-Indexdatei ein Flag („isCompared“) zugeordnet.

Wenn alle Dateien verglichen wurden, wird zwischen einer weichen Synchronisierung und einer harten Synchronisierung unterschieden. Diese Entscheidung obliegt dem Benutzer. Bei der weichen Synchronisierung werden nur die Elemente aus dem Quellverzeichnis ins Zielverzeichnis kopiert, die nicht im Ziel vorhanden sind.

Bei der harten Synchronisierung werden alle Elemente aus dem Zielverzeichnis gelöscht, die nicht auch im Quellverzeichnis vorhanden sind. Das Zielverzeichnis wird also 1:1 auf den Stand des Quellverzeichnisses gebracht. Im Quellcode wird dies über eine Schleife realisiert. In dieser werden alle Elemente in der Ziel-Indexdatei mit der Quell-Indexdatei verglichen, so lange sie noch nicht in der vorherigen Schleife verglichen wurde (Flag „isCompared“). Es werden also nun alle Elemente gesucht, die nur in der Quelle oder nur im Ziel auftauchen. Sie werden unter Angabe ihres absoluten Pfades in einer weiteren neuen Liste gespeichert. Mittels dieser werden die Elemente später gelöscht, da sie nicht im Quellverzeichnis vorgekommen sind.

5.3 Synchronisierung

Die Synchronisierung nutzt nun die beiden Listen, die die zu kopierenden bzw. die zu löschenden Dateien beinhaltet. Mithilfe einer Schleife wird die erste Liste abgearbeitet. Alle angegebenen Dateien werden so in das entsprechende Verzeichnis kopiert. Wenn die Schleife abgeschlossen ist, wird eine weitere Schleife angestoßen. In ihr werden sequenziell alle Elemente im Zielverzeichnis löscht, die nicht auch im Quellverzeichnis vorhanden sind.

5.4 FTP-Verbindungen

Bei einer Synchronisierung über das „File Transfer Protocol“ verändern sich nur die Funktionen „Analyse“ und „Synchronisierung“. Der Grund hierfür liegt darin, dass erst eine FTP-Verbindung vom Quell- zum Zielclient aufgebaut und gegebenenfalls Anmeldedaten erfasst werden müssen.

5.5 Daemon-Betrieb

Die Wunschanforderung eines Daemon-Betriebs soll das automatische Synchronisieren ermöglichen. Dafür soll ein Verzeichnis dauerhaft überprüft werden, welches in einem vom Nutzer definierten Zeitintervall mit einem Anderen synchronisiert wird.

6 Benutzeroberfläche

Die nachfolgenden Abbildungen stellen grobe Vorstellungen der Benutzeroberfläche der Anwendung dar. Die Abbildungen zeigen noch nicht die finale Benutzeroberfläche. Sie dienen der Eindrucksvermittlung der Interaktionsmöglichkeiten des Nutzers mit dem Programm. Einige Funktionen sind darin noch nicht so enthalten wie sie später tatsächlich verbaut werden, wie zum Beispiel das Anzeigen der Vergleichsdatei.

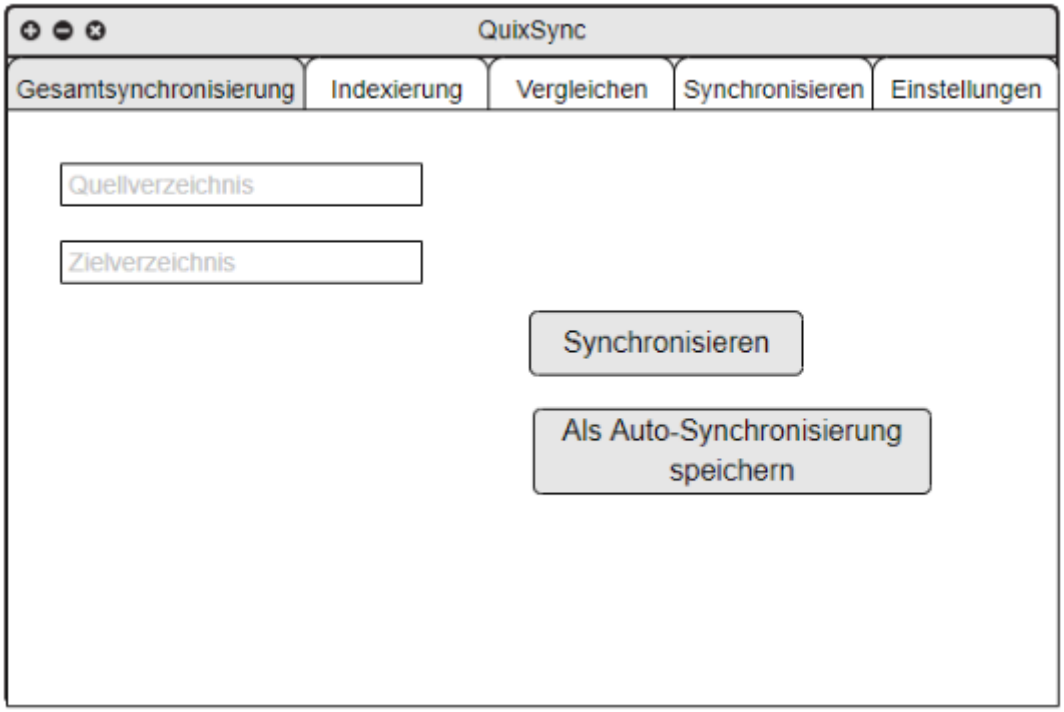

Abbildung 10 Hauptseite

Auf der Hauptseite der graphischen Benutzeroberfläche wird die Gesamtsynchronisierung angeboten, bei der alle notwendigen Schritte nacheinander ablaufen. Dazu müssen die beiden zu synchronisierenden Verzeichnisse angegeben werden. Ebenfalls wird hier die Konfiguration der Wunschanforderung einer automatisierten Synchronisierung (Daemon), unter der zusätzlichen Angabe eines Zeitintervalls, stattfinden.

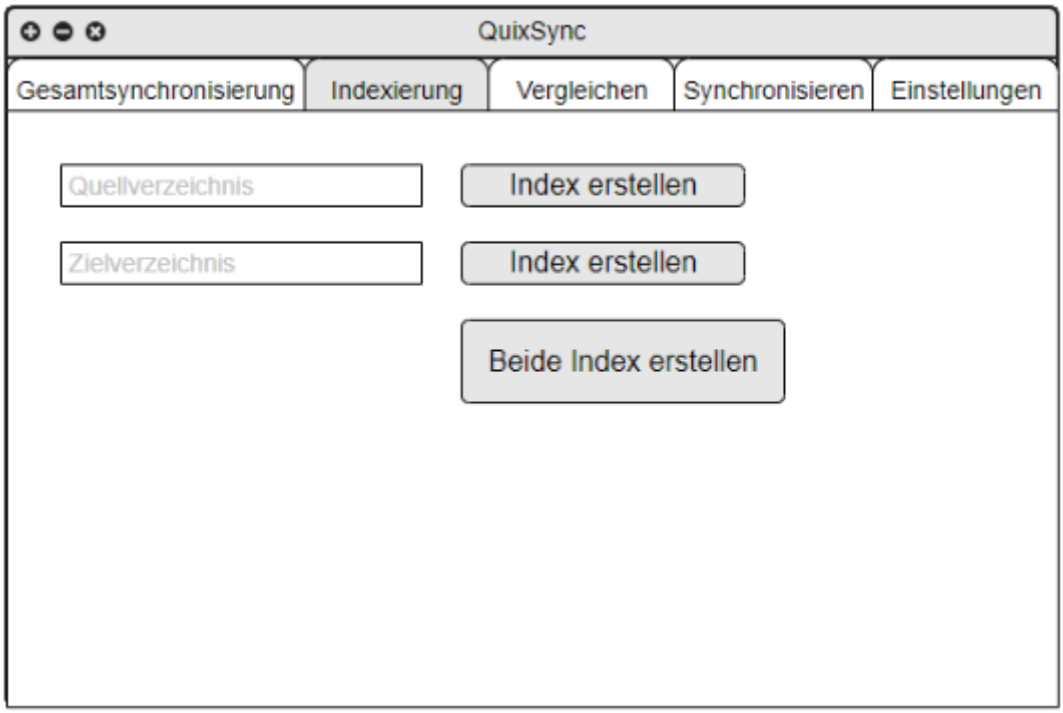

Abbildung 11 Tab „Indexierung“

Auf dieser „Seite“ der Anwendung ist der Benutzer in der Lage manuell eine Index-Datei des Quell- bzw. des Zielverzeichnisses zu erzeugen. Sie werden in einem temporären Verzeichnis gespeichert.

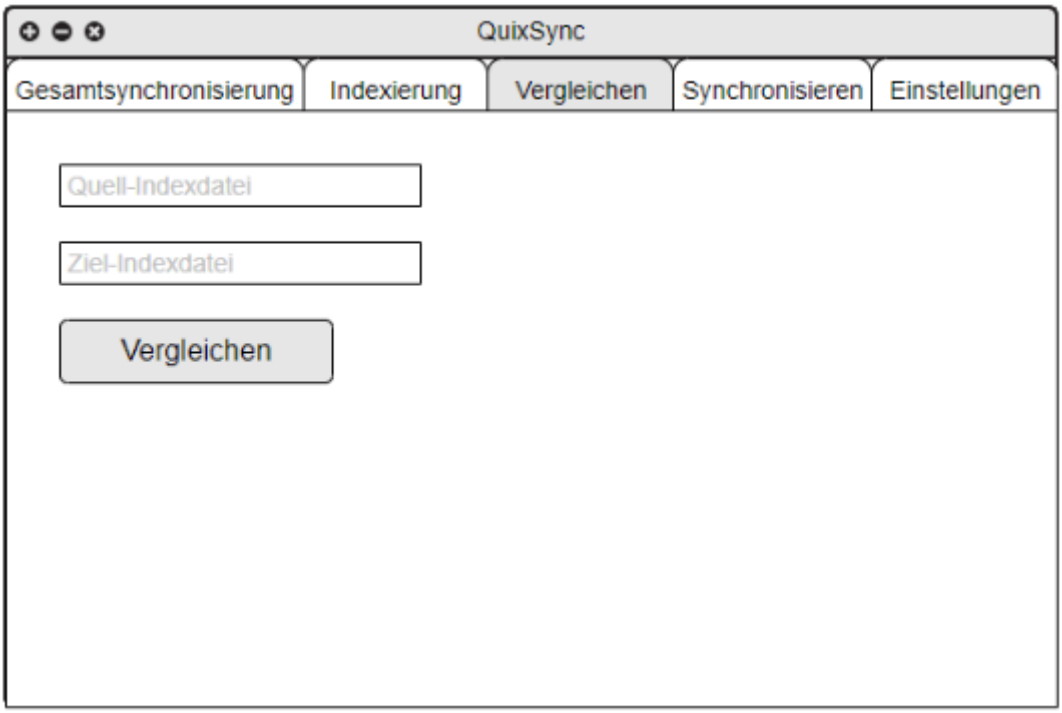

Abbildung 12 Tab „Vergleichen“

Im Unterpunkt „Vergleichen“ kann eine Vergleichsdatei erstellt werden. Sie wird aus dem Vergleich der zwei Indexdateien erzeugt und wird ebenfalls in ein temporäres Verzeichnis abgelegt. Die Vergleichsdatei soll anschließend an dieser Stelle vereinfacht präsentiert werden. Jedoch wurde diese Anzeige in dieser Vorschau noch nicht mit realisiert.

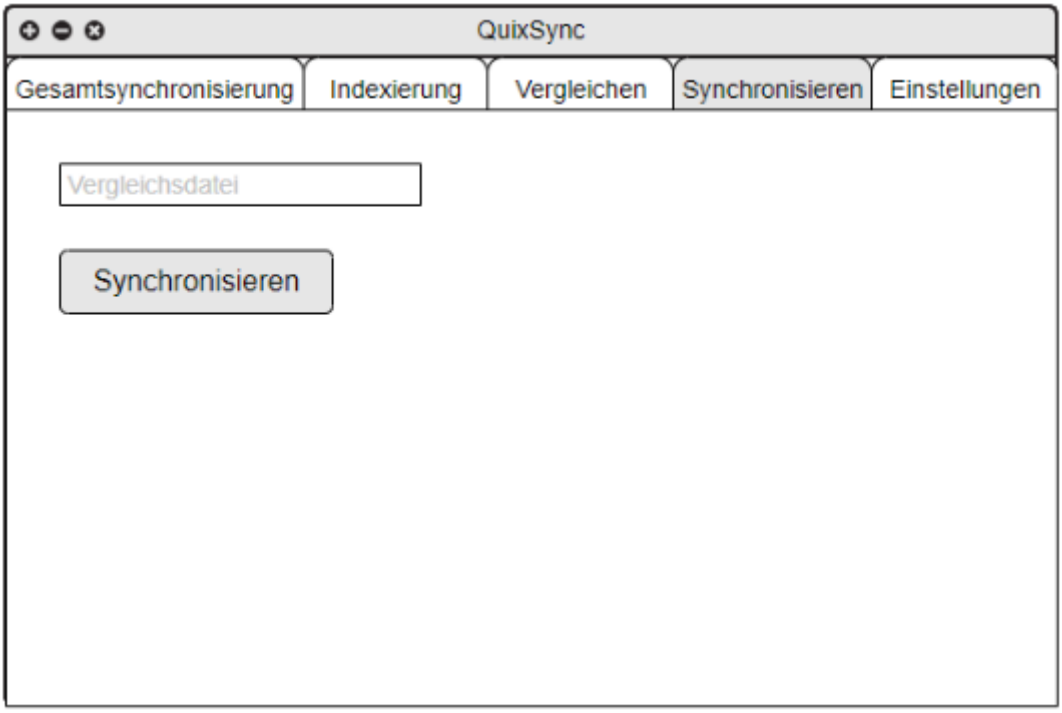

Abbildung 13 Tab „Synchronisierung“

An dieser Stelle wird dem Programm eine Vergleichsdatei übergeben. Mithilfe dieser wird das Programm eine Synchronisierung anstoßen und in einer Prozent-Anzeige den Fortschritt darstellen.

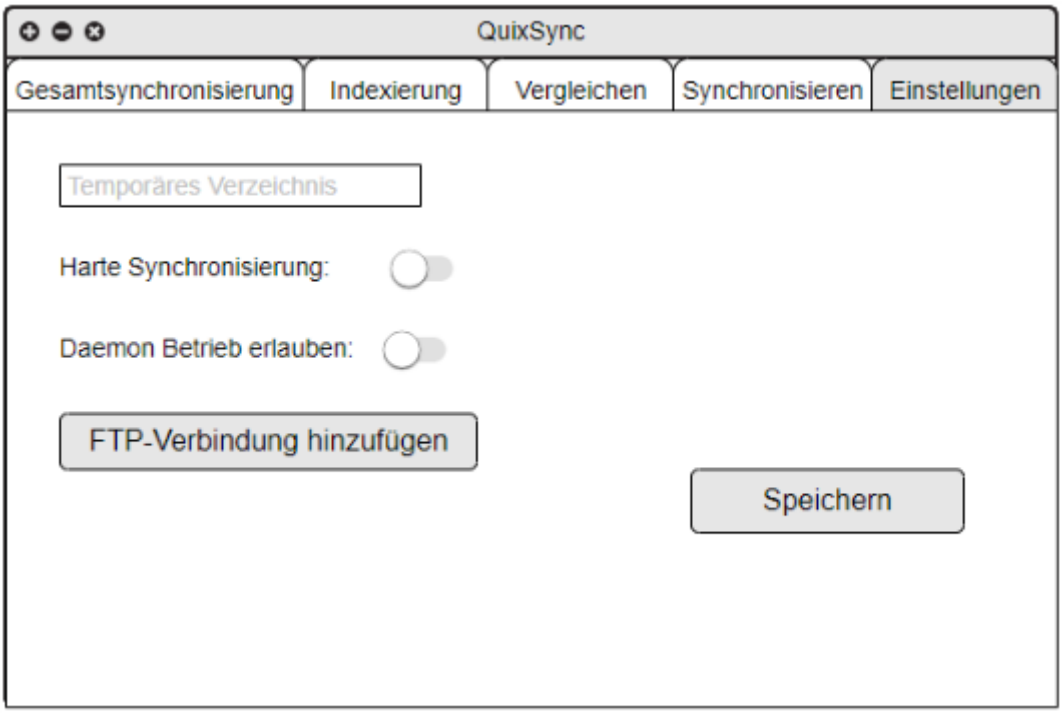

Abbildung 14 Tab „Einstellungen“

In den Einstellungen hat der Nutzer die Möglichkeit das temporäre Verzeichnis verändern. Ebenfalls kann hier zu einer harten Synchronisierung umgeschaltet werden, eine FTP-Verbindung hinzugefügt werden, wie auch der Daemon-Betrieb erlaubt werden. Durch Letzteres wird der Button Auto-Synchronisierung in Abbildung 1 aktiv geschalten.

7. Abbildungsverzeichnis