

Realisierungsbericht

Niklaus Hofer, Lukas Knöpfel, Kaleb Tschabold

May 10, 2011

Status	In Arbeit/ In Prüfung / Abgeschlossen
Projektname	Projektexplorer
Projektleiter	Lukas Knöpfel
Auftraggeber	M. Frieden, GIBB
Autoren	Kaleb Tschabold, Lukas Knöpfel, Niklaus Hofer
Verteiler	Lukas Knöpfel, Kaleb Tschabold, Niklaus Hofer

Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle
0.1	08.02.2011	Gesammelten Text einfügen	Kaleb Tschabold
0.9	08.02.2011	Abgabebereit	Kaleb Tschabold
0.99	May 10, 2011	Transfer nach \LaTeX	Niklaus Hofer
1.0	May 10, 2011	Korrekturen	Niklaus Hofer

Definitionen und Abkürzungen

Begriff/ Abkürzung	Bedeutung
CLI	Command Line Interface

Contents

1	Zweck des Dokuments	5
2	Technische Detailspezifikation	5
2.1	Innere Struktur	5
2.1.1	Lösungsvorschläge für die Struktur des Systemdesigns	5
2.1.1.1	GUI	5
2.1.1.2	Datenstruktur	5
2.1.2	Struktur des Systemdesigns	5
2.1.3	Beschreibung der Elemente	5
2.2	Schnittstellendefinition	6
2.3	Datenmodell	6
2.3.1	Datenbank	6
2.3.2	File-object	6
2.4	Sicherheit	7
2.5	Anforderungszuordnung	7
3	Systemdokumentation	8
3.1	Inline-Dokumentation	8
3.2	Benutzerhandbuch	8
3.2.1	Systemübersicht	8
3.2.1.1	Aufgabengebiet des Programms	8
3.2.1.2	Programmoberfläche	8
3.2.1.3	Anmerkungen zur korrekten Verwendung unter dem Aspekt der Sicherheit	9
3.2.2	Anwenderfunktionalität	9
3.2.2.1	Ansicht wechseln	9
3.2.2.2	Verzeichnis wechseln	10
3.2.2.3	Hisotry	10
3.2.2.4	Dateien Öffnen	10
3.2.2.5	Tags hinzufügen	10
3.2.2.6	Tags entfernen	10
3.2.2.7	Dateien anhand der Tags durchsuchen	10
3.2.2.8	Fehlermeldungen	10
3.3	Supporthandbuch	11
3.3.1	Massnahmen bei Benutzerproblemen	11
3.3.2	Massnahmen bei technischen Problemen	11
3.3.3	Anhang zum Supporthandbuch	11
4	Systemtest	11
4.1	Testspezifikation	11
4.1.1	Kritikalität der Funktionseinheit	11
4.1.2	Testanforderungen	11
4.1.3	Testverfahren	11
4.1.4	Testkriterine	11
4.1.5	Testfälle	11
4.2	Testprozedur	11
4.2.1	Vorbereitung	11
4.2.2	Durchführung	11
4.2.3	Nachbearbeitung	11
4.3	Testprotokoll	11
4.3.1	Testobjekt	11
4.3.2	Testresultate	11
4.3.3	Testauswertung	11
5	Mittelbedarf	11
6	Planung und Organisation	11
7	Wirtschaftlichkeit	11

8	Konsequenzen	11
9	Antrag auf Freigabe der nächsten Projektphase	11

1 Zweck des Dokuments

Wir hatten jetzt einige Wochen Zeit um an der Realisierung zu arbeiten. Wir konnten jetzt unsere Programmspezifikationen noch genauer ausarbeiten, weil wir während dem Programmieren gesehen haben was noch verbessert oder ergänzt werden sollte. In diesem Dokument sind jetzt die genauen Informationen zum Programm.

2 Technische Detailspezifikation

2.1 Innere Struktur

2.1.1 Lösungsvorschläge für die Struktur des Systemdesigns

Es gibt zwei wichtige Entscheidungen zum Systemdesign, die während der Realisierung getroffen wurden. Die Erste betrifft, das GUI, die zweite die Art wie die Daten in der Datenbank abgelegt werden.

2.1.1.1 GUI Die Änderung am GUI betrifft die Art und Weise wie die Tags zu den Dateien zugeordnet werden. Unser erster Einfall dazu war der, dass sich über das Kontextmenü der Dateien ein Popup öffnen liesse, in dem die Tags zugeordnet werden könnten.

Für ein Programm, dessen Hauptaufgabe gerade die Verwaltung der Tags darstellt, ist diese Art Tags Dateien zu ordnen aber recht aufwendig.

Die Verwaltung der Tags wird nun unabhängig von der Ansicht (Tag oder Hierarchisch) immer auf der rechten Seite des Programms angezeigt. Sobald in der linken Spalte eine Datei angewählt wird, werden deren Tags in der rechten aufgelistet.

Zudem können der Dateien von dort aus weitere, bereits bestehende, Tags per Doppelklick zugeordnet oder ganz neue hinzugefügt werden, indem man deren Namen, Komma getrennt, der Liste der Tags anhängt.

Diese Lösung, für die wir uns entschieden haben ist weniger umständlich und macht die Aufgabe des Programms gleich beim Start deutlich.

2.1.1.2 Datenstruktur Die zweite wichtige Entscheidung betrifft die Art, wie die Pfade zu den Dateien in der Datenbank abgelegt werden.

Damit Dateien anhand ihrer URI in der Datenbank gefunden werden können muss die Art, wie die URI abgelegt wird immer gleich sein.

Aus Datenbank-technischer Sicht ist es von Vorteil, den Pfad zur Datei vom Dateinamen getrennt zu speichern. Abfragen nach 'allen Dateien aus dem Verzeichnis X' werden so deutlich einfacher auszuführen.

Auch beim Darstellen der Dateien ist diese Art des Speicherns meist von Vorteil, da der Dateiname immer getrennt vom Pfad dargestellt wird (der Pfad oben, wie von Windows Explorer gewöhnt, und der Dateiname unten im mittleren Panel).

Diese getrennte Speicherung hat aber zu der Frage geführt ob / (oder in Windows) am Ende der Pfadangabe mitgespeichert werden sollte.

Wir entschieden uns dafür, damit Pfade ohne weiteren Aufwand vollständig zusammengesetzt werden können.

Ist der 'Dateiname' der Name eines neuen Verzeichnisses, so trägt auch er ein / (oder in Windows) am Ende.

2.1.2 Struktur des Systemdesigns

2.1.3 Beschreibung der Elemente

Main Wird zum Starten des Programmes aufgerufen. Main.py instanziiert alle weiteren Elemente die für das Funktionieren des Programms nötig sind und koordiniert die Kommunikation zwischen den einzelnen Elementen.

DB DB.py ist ein Interface zur Datenbank. Es nimmt allen anderen Klassen die Aufgabe ab selbst SQL statements ab zu setzen und bietet stattdessen nach aussen hin verschiedene Funktionen an um Daten zu lesen oder zu schreiben.

Utility Enthält verschiedene nützliche Methoden die immer mal wieder von einzelnen Teilen des Programmes benötigt werden.

CLI Dient als Command-line-interface für das Programm. Es nimmt über die Kommandozeile beim Aufruf verschiedene Befehle entgegen die es dann ausführt.

TagManager Der Tag Manager ist für kleine Tag Verwaltungsaufgaben zuständig.

FileManager Über den FileManager wird auf das Dateisystem zugegriffen. Hier wird aus jeder Datei aus dem File System ein File Objekt erstellt.

FileSystemListener Registriert beim Kernel Listener für zu überwachende Ordner. Wird in diesen Ordnern eine Operation ausgeführt (wie das Verschieben, Löschen, Umbenennen oder Erstellen einer Datei), so wird der FileSystemListener vom Kernel darüber in Kenntnis gesetzt, woraufhin er wiederum die nötigen Aktionen auslöst um die Datenbank auf dem aktuellen Stand zu halten.

Dies soll dazu beitragen, dass möglichst selten Dateien angezeigt werden, die auf Dateisystem-Ebene nicht existieren.

GUI Ist für die grafische Darstellung des Programms mittels GTK zuständig. Das GUI ist in der Lage je nach Bedarf eine andere 'View' darzustellen. Direkt nach dem Programmstart wird HierarchicalView dargestellt. Im Betrieb kann jederzeit zwischen 'TagView' und 'HierarchicalView' umgeschaltet werden. Dazu kann GUI, per Polymorphismus, eine Klasse aufnehmen die von 'View' erbt.

TagView Enthält die Darstellung der Tag-Ansicht und wird von 'GUI' bei Bedarf geladen.

HierarchicalView Enthält die Darstellung der hierarchischen Ansicht und wird von 'GUI' bei Bedarf geladen.

View Mutterklasse von GUI.

File Repräsentiert eine Datei und wird benutzt um Informationen über Dateien zwischen den Elementen des Programms auszutauschen. Für mehr Informationen siehe 2.3.2.

2.2 Schnittstellendefinition

1. Interne Schnittstellen

- a) Intern ist die Kommunikation mit der Datenbank sehr wichtig.
 - i. Die Datenbank-Schnittstelle ist in DB.py implementiert.
 - ii. Die Schnittstelle nimmt in den meisten Fällen Objekte vom Typ File(.py). In anderen auch Strings.
 - iii. Welche Funktionen in der Schnittstelle genau definiert sind, kann dem Klassendiagramm entnommen werden.
Wie die einzelnen Methoden aufzurufen sind und was sie genau tun kann, ist jeweils im Methodenkommentar ersichtlich.

2. Externe Schnittstellen

- a) Für den normalen Gebrauch haben wir das GUI. Beim GUI wird Wert auf das einfache Verwalten von Tags und Dateien gelegt.
 - i. Das GUI bietet zwei Modi, einer der den herkömmlichen Dateimanagern mit hierarchischer Ansicht entspricht und
 - ii. Einen Tagmodus, in dem sich Dateien anhand deren Tags durchsuchen und ordnen lassen.
- b) Für scripting oder für solche Systeme ohne grafische Ausgabe haben wir eine CLI Version. Es wird besonders Wert auf das einfache Aufrufen von anderen Programmen (Scripts) gelegt. (nicht implementiert)
 - i. Die Kommandos sollen in der Bedienung weitgehend mit dem standard Unix-Tools kompatibel

2.3 Datenmodell

2.3.1 Datenbank

Die Datenbank lässt sich über DB.py ansprechen, siehe Schnittstellendefinition für mehr Informationen. Welche Felder genau welche Information enthalten ist im folgenden Abschnitt erläutert.

2.3.2 File-object

Informationen über einzelne Dateien werden innerhalb des Programms mit Hilfe des File objects festgehalten und ausgetauscht.

Das File Object hat für alle Variablen Getter und Setter, sie alle können aber auch im Konstruktor angegeben werden.

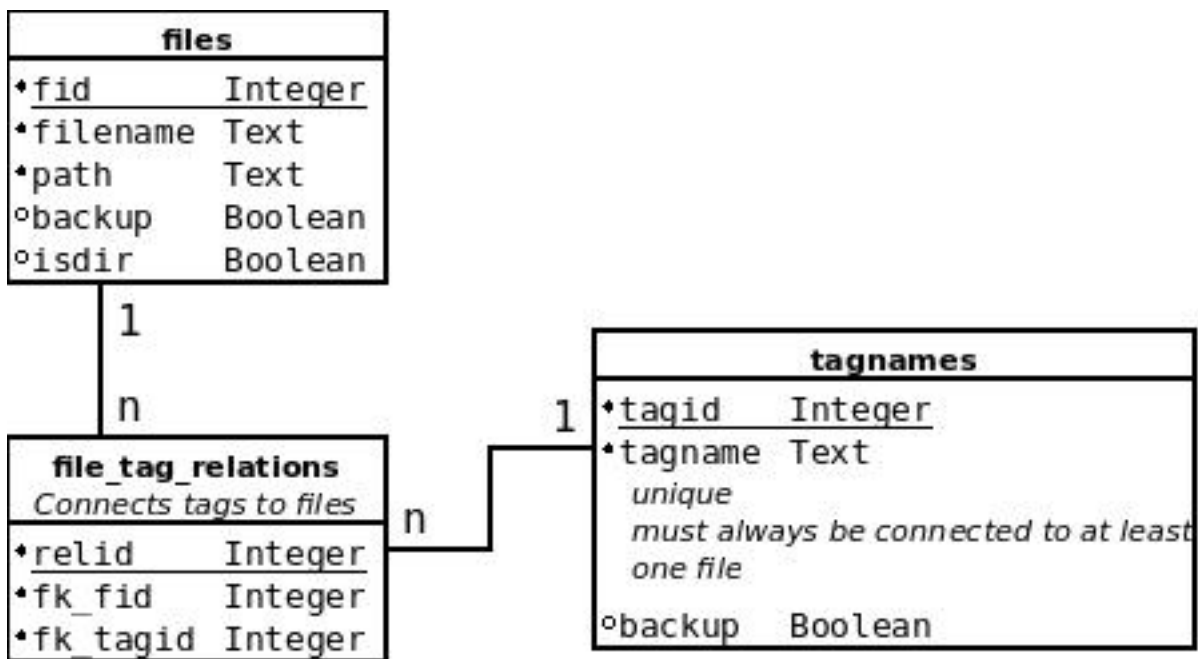


Figure 1: Datenbankschema

Hier eine Auflistung und Erläuterung zu den einzelnen Variablen der File Klasse:

Name	Type	Erläuterungen	Entsprechung in der Datenbank
fileName	String	Hält den Namen der Datei. Ist der 'Name' der eines Verzeichnisses, so endet er mit / (oder \in Windows)	files.filename
path	String	Hält den Pfad zu dem Verzeichnis in dem die Datei liegt. Endet mit / (oder \in Windows)	files.path
isDir	Boolean	Besagt, ob das Objekt eine Datei oder ein Verzeichnis repräsentiert.	files.isdir
backup	Boolean	Besagt, ob Backups der Datei angelegt werden sollen.	files.backup
fullPath	String	Der ganze Pfad zur Datei inklusive deren Namen. Wird der Fullpath an ein File object übergeben, so wird dieser mittels Regex zerlegt und die Werte in fileName und path abgelegt.	files.path + files.filename
tags	list[String]	Enthält eine Liste aller Tags die der Datei zugeordnet sind als Strings.	Die Werte in Tagnames werden mit file_tag_relation mit denen in files verknüpft.

2.4 Sicherheit

Die Datenbank liegt auf dem lokalen Dateisystem und stellt nach Aussen (über das Netzwerk) keine Schnittstelle zur Verfügung.

Für jeden Nutzer der Software wird in dessen Home-Verzeichnis (unter Unix also /home/username/) ein Ordner ./project-browser angelegt, in dem sich die Datenbank-Datei befindet.

Die Datenbank ist also durch die Zugriffsberechtigung des Dateisystems geschützt und fügt sich somit nahtlos in bestehende Sicherheits- und Datenschutzkonzepte ein.

2.5 Anforderungszuordnung

1. Main
2. DB
3. Utility

4. CLI
5. TagManager
6. FileManager
7. FileSystemListener
8. GUI
9. TagView
10. HierarchicalView
11. View
12. File

Nr.	Anforderungen	1	2	3	4	5	6	7	8	9	10	11	12
1	Das Programm starten												
2	GUI vorhanden												
3	CLI vorhanden												
4	Tag hinzufügen												
5	Datei auswählen												
6	Versionierung												
7	GUI: Versionierung für Tags												
8	Dateiänderungen werden erkannt												
9	Löschen wird erkannt												
10	Neue Dateien werden erkannt												
11	Hierarchische Anzeige												
12	Tag Anzeige												

3 Systemdokumentation

3.1 Inline-Dokumentation

3.2 Benutzerhandbuch

3.2.1 Systemübersicht

3.2.1.1 Aufgabengebiet des Programms Das Programm Project-Explorer ist dazu da bei der Verwaltung von Dateien zu helfen.

In einer gewöhnlichen Arbeitsumgebung werden Dateien in einem hierarchischen System aus Ordnern abgelegt. Doch mit der Datenmenge steigt auch die Komplexität dieser hierarchischen Strukturen und besonders Dateien die selten verwendet werden können schwierig aufzufinden sein.

Programme zum Verwalten von Musik- und Bilddateien bieten deshalb schon seit vielen Jahren zusätzlich sogenannte Metadaten an, um weitere Informationen zu der Datei (z.B. Album, Artist, Aufnahmejahr, ... bei Musikdateien) zu speichern mit deren Hilfe sich diese dann einfacher wiederfinden lassen.

Projekt-Explorer hat das Ziel ähnliches anzubieten - aber nicht auf eine Art von Dateien beschränkt, sondern für jede Datei, die auf dem lokalen Rechner liegt.

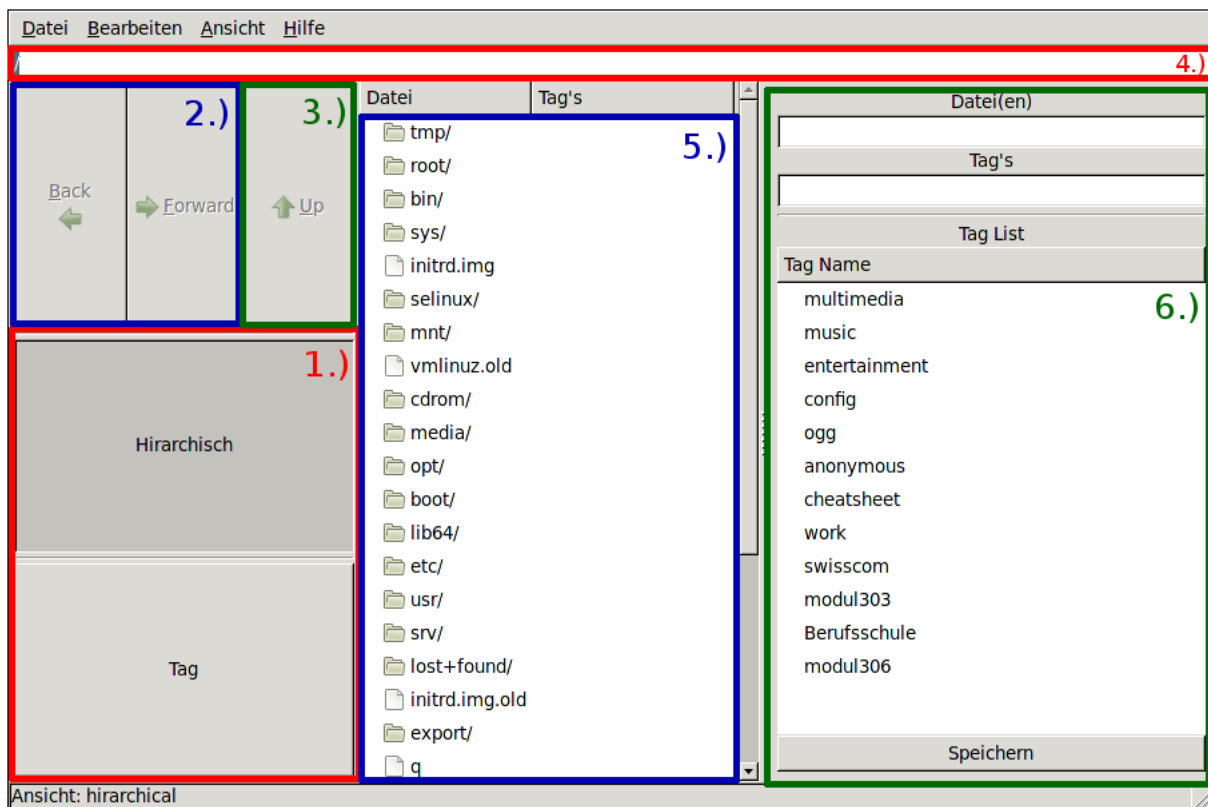
Der Nutzer kann dazu den einzelnen Dateien Tags zuordnen, anhand derer er die Dateien später wieder finden kann. Die Tags können selbst festgelegt und vergeben werden.

3.2.1.2 Programmoberfläche Die Oberfläche des Programms lässt sich grob in vier Bereiche aufteilen.

Der Erste (4. auf dem Bild), ist die Multibar. Hier kann der Pfad zu einem Verzeichnis eingegeben werden, oder, in der Tag-Ansicht, ein Tag nachdem man sucht.

Der Zweite Bereich (1., 2. und 3. auf dem Bild) ist der Navigationsbereich. Hier kann in der Ordnerstruktur navigiert und zwischen den Ansichten umgeschaltet werden. Im dritten Bereich (5. auf dem Bild), werden die Dateien angezeigt. Rechts der Dateien sind deren Tags zu sehen.

Der letzte Bereich (6. auf dem Bild) ist der wichtigste. Hier werden die verfügbaren Tags angezeigt und hier können die Tags den Dateien zugeordnet werden.



3.2.1.3 Anmerkungen zur korrekten Verwendung unter dem Aspekt der Sicherheit Den Autoren ist es wichtig an dieser Stelle einige Bemerkungen zur Datensicherheit zu machen. Grundsätzlich gilt für die Daten die im Projekt-Explorer erfasst werden dasselbe wie für alle anderen Daten auf dem Computer auch. Die Informationen werden in einem persönlichen Verzeichnis des Nutzers abgelegt.

Per Standard ist dieses vor dem Zugriff durch andere Benutzer geschützt. Es kann aber in einzelnen Fällen sein, dass diese Einstellung vom Systemadministrator verändert worden ist. Es ist zudem zu beachten, dass der Systemadministrator zu jeder Zeit Zugriff auf die Daten hat. Werden im Program sehr persönliche Informationen abgelegt oder handelt es sich beim verwendeten Gerät um einen portablen Computer (Ein Laptop oder gar ein Handset) so sollte die lokale Festplatte verschlüsselt werden, damit im Falle eines Verlustes des Gerätes kein unberechtigter Zugriff auf die Dateien geschehen kann.

3.2.2 Anwenderfunktionalität

In dieser Kategorie werden wir einige typische Anwendungen für Projekt-Explorer beschreiben. Die Erläuterungen werden sich zumeiste auf das Bild im Abschnitt 'Programmoberfläche' beziehen. Nummern aus dem Bild werden im Format #1.) angegeben. Wir werden folgende Szenarien beschreiben:

- Wechseln der Ansicht zwischen Tag- und Hierarchischer Ansicht.
- Wechseln in ein anderes Verzeichnis und Nutzen der Autovervollständigung.
- Navigieren in der History.
- Öffnen von Dateien.
- Tags zu Dateien zuordnen.
- Tags von Dateien entfernen.
- Dateien eines Tags anzeigen.
- Fehlermeldungen

3.2.2.1 Ansicht wechseln Projekt-Explorer hat zwei verschiedene Ansichten, nämlich 'Hierarchisch' und 'Tags'. Beim Wechseln zwischen den Ansichten ändert sich die Ansicht #5.).

Die Ansicht kann entweder über die grossen Knöpfe in #1.) gewechselt werden oder in der Menüleiste unter dem Punkt "Ansicht".

3.2.2.2 Verzeichnis wechseln Um in der hierarchischen Ansicht das Verzeichnis zu wechseln, gibt man das gewünschte Zielverzeichnis in #4.) ein. Während der Eingabe erscheint ein Drop-Down mit Verzeichnisschlägen das während des Tippens laufend aktualisiert wird. In diesem Drop-Down kann ein Verzeichnis mittels der Pfeiltasten auf der Tastatur und Enter, oder mit der Maus ausgewählt werden.

Um in das Verzeichnis oberhalb zu wechseln, kann der Knopf "UP" in #3.) geklickt werden. Um in ein Unterverzeichnis zu öffnen, führt man einen Doppelklick auf den entsprechenden Eintrag in #5.) aus.

3.2.2.3 History Projekt-Explorer merkt sich in welchem Verzeichnis man zuletzt war. Diese Werte werden zur Laufzeit in der History gespeichert. Über die Buttons "Back" und "Forward" kann in dieser History navigiert werden. Die History von Projekt-Explorer entspricht vom Konzept her ziemlich genau der entsprechenden Funktionalität von Webbrowsern.

Beim Schliessen der Applikation geht die History verloren.

3.2.2.4 Dateien Öffnen Dateien werden durch einen Doppelklick geöffnet. Sie werden mit dem Programm geöffnet, das auf dem Betriebssystem als Standardapplikation für den entsprechenden Dateityp festgelegt ist. Ist für den Dateityp keine Standard-Applikation festgelegt, so kann die Datei auch nicht geöffnet werden. Wichtig! Klickt man in der Tagview auf einen Ordner, so wechselt die Ansicht automatisch zur hierarchischen Ansicht zurück.

3.2.2.5 Tags hinzufügen Um einer Datei Tags hinzu zu fügen muss diese erst in #5.) angewählt werden. Das geschieht über einmaliges Klicken auf die Datei. Anschliessend werden in #6.) Informationen zu der Datei angezeigt.

Im oberen Feld der Name, im unteren Feld die Tags. Die einzelnen Tags sind jeweils durch ein Komma (",") getrennt.

Im unteren Bereich von #6.) sind alle bereits verwendeten Tags ersichtlich. Um der Datei eines dieser Tags hinzu zu fügen, doppelklickt man den Eintrag, woraufhin dieser in der Liste oben erscheint. Alternativ dazu, kann das Tag auch von Hand in der Liste geschrieben werden.

Um die Änderung zu übernehmen klickt man auf den Knopf 'speichern', der sich ganz unten in #6.) befindet.

Will man einen gänzlich neuen Tag anfügen, so muss dieser von Hand in die Liste der Tags geschrieben werden.

Natürlich muss er durch ein Komma von den anderen Tags getrennt sein. Nach dem speichern, erscheint der neue Tag auch in der Liste unten.

Wichtig! Es können keine Tags bestehen, die keiner Datei zugeordnet sind!

3.2.2.6 Tags entfernen Um ein Tag von einer Datei zu entfernen, löscht man den Eintrag aus der Liste 'Tags' in #6.) und drückt speichern.

Ist der Tag mit keiner weiteren Datei verbunden, so verschwindet er aus der Auswahlliste in #6.).

3.2.2.7 Dateien anhand der Tags durchsuchen Um Dateien eines Tags anzuzeigen wechselt man erst in die Tagview. Alle Tags werden dort übereinander angezeigt. Durch das Doppelklicken eines Eintrages werden alle Dateien dieses Tags sichtbar.

Alternativ kann der Tagname auch in #4.) eingegeben werden. Während der Eingabe werden Tags vorgeschlagen, die gleich geschrieben werden und im Drop-Down zur Auswahl gestellt.

3.2.2.8 Fehlermeldungen Im Normalbetrieb werden keine Fehlermeldungen ausgegeben. Um Trotzdem Informationen zu auftretenden Fehlern zu erhalten muss das Programm aus dem Terminal gestartet werden. Auftretende Fehler werden dann dort ausgegeben.

3.3 Supporthandbuch

3.3.1 Massnahmen bei Benutzerproblemen

- Die Aufgabenstellung von Projekt-Explorer ist lediglich die Verwaltung der Tags. Obwohl im Programm selbst sehr wohl Dateien angezeigt werden, können Dateien weder mit copy 'nd paste noch per drag 'nd drop in den Ordnern bewegt werden. Das Programm selbst bietet auch keine Möglichkeit zum Umbenennen oder Löschen von Dateien.
- Es ist darauf zu achten, dass die Grösse der beiden rechten Pannels von Projekt-Explorer frei verstellbar ist, indem man die 'Trennlinie' mit der Maus weiter nach links oder rechts verschiebt. Schiebt man diese Linie zu weit in die eine Richtung kann es vorkommen, dass eine der Spalten komplett verschwindet. Das lässt sich ganz einfach beheben, indem man die Linie wieder verschiebt.

- Klickt man einen der Buttons zum Umschalten der Ansicht auf der linken Seite des Programmfensters mehrmals hintereinander, so ändert die Ansicht wiederholt.

3.3.2 Massnahmen bei technischen Problemen

- Lässt sich das Programm nicht starten, so ist sicher zu stellen, dass sowohl Python 2.7.X als auch das mitgelieferte PyGTK korrekt installiert sind.
- Werden Daten falsch angezeigt, so ist zu befürchten, dass in der Datenbank korrupte Daten liegen. Es gibt zwei Wege dies zu beheben:
 - Man kann die Datenbank manuell mit einem beliebigen SQLite Programm öffnen, die korrupten Einträge suchen und Löschen.
Diese Methode ist vorzuziehen, da so alle Daten erhalten bleiben.
Der Pfad zur Datenbank ist `/.project-explorer/db`.
 - Die zweite Lösung besteht darin, die Datenbank einfach zu Löschen. Beim nächsten Start erstellt das Programm dann automatisch eine neue, leere Datenbank.
Bei dieser Methode gehen alle Daten verloren!
Der Pfad zur Datenbank ist `/.project-explorer/db`

3.3.3 Anhang zum Supporthandbuch

4 Systemtest

4.1 Testspezifikation

4.1.1 Kritikalität der Funktionseinheit

4.1.2 Testanforderungen

4.1.3 Testverfahren

4.1.4 Testkriterine

4.1.5 Testfälle

4.2 Testprozedur

4.2.1 Vorbereitung

4.2.2 Durchführung

4.2.3 Nachbearbeitung

4.3 Testprotokoll

4.3.1 Testobjekt

4.3.2 Testresultate

4.3.3 Testauswertung

5 Mittelbedarf

6 Planung und Organisation

7 Wirtschaftlichkeit

8 Konsequenzen

9 Antrag auf Freigabe der nächsten Projektphase