

Seminararbeit

Advanced Topics in iOS

Thema: Annotieren von PDFs

Author

Technische Universität München
Fakultät für Informatik

1 Themenvorstellung

Inhalt dieser Arbeit ist die Implementierung eines Programms (einer App) für das iPad mit dem es möglich ist PDF Dateien inklusive Annotationen anzuzeigen und mit Annotationen zu versehen. Dabei ist es für den Benutzer möglich, dass er PDF Dateien über iTunes auf sein iPad lädt. Es können einzelne Wörter und ganze Absätze umrahmt werden und Annotationen mithilfe der Tastatur hinzugefügt werden. Diese Markierungen und Annotationen werden abgespeichert und können bei erneutem öffnen der selben Datei auch wieder entfernt werden.

Teil dieser Arbeit ist auch eine Code Dokumentation, welche zusammen mit dem Code als HTML in einem Git-Repository abgespeichert ist.

2 Realisierungsmöglichkeiten

Aus meiner Sicht gibt es zwei verschiedene Möglichkeiten wie dieses Thema umgesetzt werden kann. Beide haben die Gemeinsamkeit, dass als erster Schritt ein PDF Viewer implementiert werden muss, da die von Apple zur Verfügung gestellten Methoden (Webview, QuickLook Framework) weder die Anzeige von bereits vorhandenen Annotationen noch die Anreicherung der PDFs mit Informationen zulassen. Dazu sollen frei verfügbare Bibliotheken eingebunden werden.

Die beiden verschiedenen Realisierungsmöglichkeiten unterscheiden sich im Wesentlichen in der Art und Weise wie Annotationen und Markierungen gespeichert werden und wieder geladen werden.

- **Internes Speichern der Informationen (in der App)**

Eine Möglichkeit die erstellten Informationen zu speichern ist, dass die Informationen separat von der PDF Datei gespeichert werden. Dazu ist eine Klasse zu erstellen, die die Informationen eindeutig einer PDF Datei zuordnen kann. Die PDF Datei wäre in dieser Klasse beispielsweise als CGPDFDocumentRef und die Annotationen beispielsweise als NSDictionary vorhanden. Die Daten werden mithilfe des CoreData Managers auf

die Platte geschrieben, so dass sich auch der Arbeitsspeicherverbrauch in Grenzen hält und eine persistente Speicherung der annotierten PDF Datei möglich ist. Das Erstellen eines Protokolls für Annotationen, das erfüllt werden muss ist auch notwendig.

- **Direktes Hinzufügen der Informationen in die PDF Datei**

Eine weitere Realisierungsmöglichkeit besteht darin, die erstellten Informationen direkt in die PDF Datei hinzuzufügen. Dazu ist es notwendig, im ersten Schritt die Struktur einer PDF Datei zu verstehen. Bei der Implementierung ist es sinnvoll auf die frei verfügbare Bibliothek PoDoFo zurückzugreifen, die verschiedene Funktionen zur Manipulation von PDF Dateien zur Verfügung stellt. Allerdings besteht hier das Problem, dass iOS nicht zu den offiziell unterstützten Plattformen gehört. Mehr dazu in *Probleme bei der Implementierung*.

Alle erstellten Informationen (Markierungen, Annotationen) werden direkt gemäß der *PDF Reference, Version 1.7*[1] in die PDF Datei gespeichert. Dadurch entsteht der Vorteil, dass die hinzugefügten Informationen nach einem Export der Datei auf ein anderes Gerät dort auch angezeigt und bearbeitet werden können.

3 Entscheidung für Direktes Hinzufügen

Ich favorisiere die zweite Methode, bei der die erstellten Informationen (Annotationen, Markierungen) direkt in die PDF Datei hinzugefügt werden. Aus dem Standpunkt des Users betrachtet ist es ein sehr großer Vorteil, wenn Informationen auch an einem Desktop Rechner betrachtet werden können. Außerdem ist die App somit allgemein besser wiederverwendbar und kann in andere Applikationen eingebunden werden, bei denen gewünscht ist die annotierten PDF Dateien per Email zu versenden. Denn dadurch kann jeder Empfänger die Annotationen lesen.

4 Umsetzung

In diesem Abschnitt werden die erstellte Benutzeroberfläche, Besonderheiten der Implementierung und die verwendeten Frameworks/Libraries erläutert.

4.1 Benutzeroberfläche

Die Benutzeroberfläche besteht aus einem SplitViewController, der nach der Initialisierung im MasterViewController alle PDF Dateien des *Shared Documents* Verzeichnisses in einem TableView anzeigt und im DetailViewController einen leeren Hintergrund anzeigt, solange keine PDF Datei ausgewählt wurde. Dieser Zustand ist in Abbildung 1 dargestellt.

Hier hat der Benutzer die Möglichkeit eine PDF Datei, die er zuvor via iTunes hinzugefügt hat auszuwählen. Die erste Seite der ausgewählten Datei wird dann im DetailViewController angezeigt. Die NavigationBar des DetailViewController zeigt nun als Titel den Namen der angezeigten Datei zusammen mit

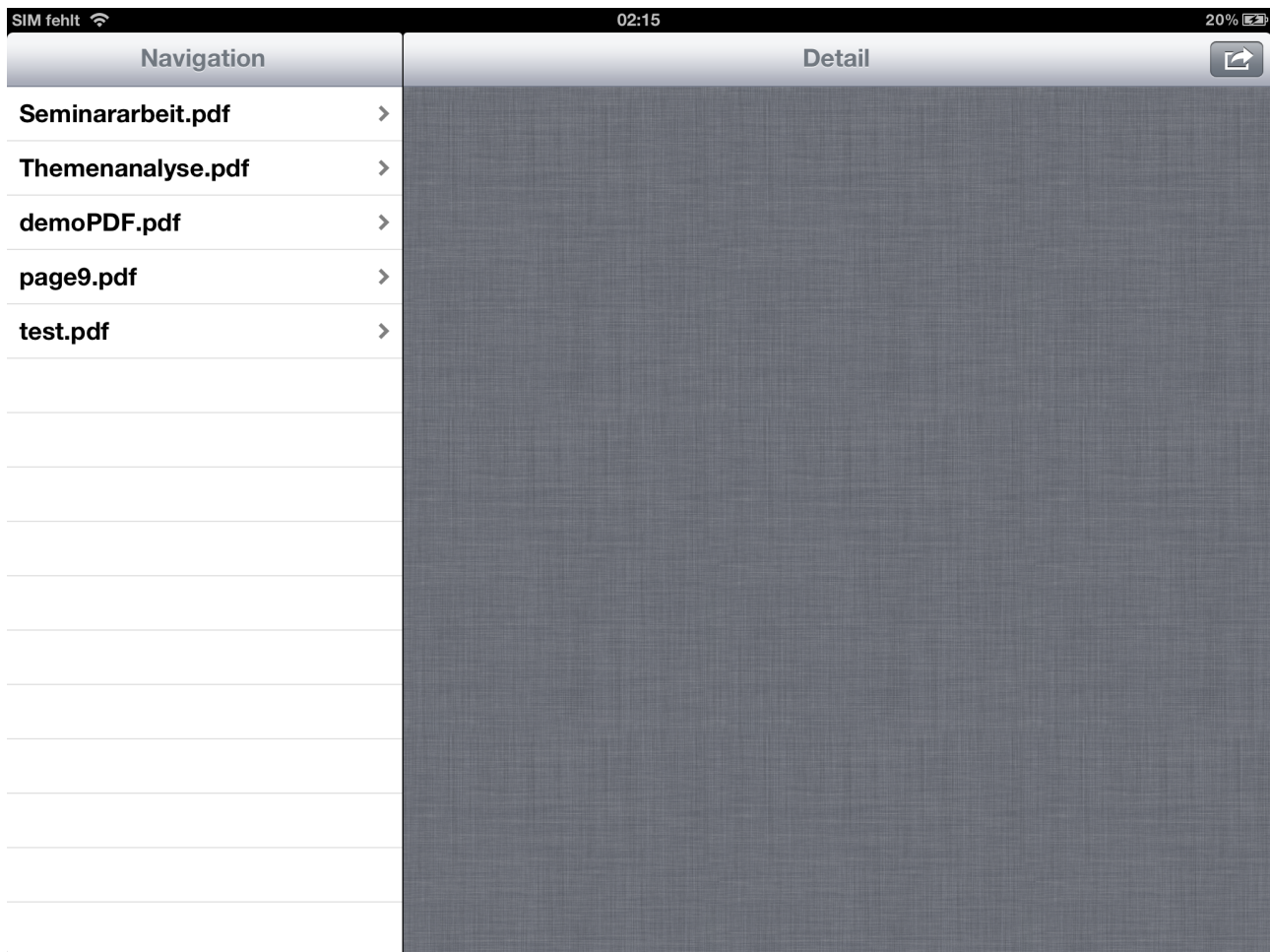


Abbildung 1: Benutzeroberfläche nach Initialisierung

der Information auf welcher Seite man sich befindet an. Im MasterViewController werden nun alle Annotationen der aktuellen Seite in einem TableView angezeigt. Dieser Zustand ist in Abbildung 2 dargestellt. Um eine Seite weiter zu blättern muss in die Mitte des rechten Randes des DetailViewControllers eine Tap Gesture ausgeführt werden. Um eine Seite zurück zu kommen muss in der Mitte des linken Randes des DetailViewControllers eine Tap Gesture ausgeführt werden.

Der Benutzer hat nun die Möglichkeit eine Annotation zu löschen. Dazu löscht er die Annotation mittels Swipe-Geste aus dem TableView im MasterViewController. Dies wird in der Abbildung 3 gezeigt. Bei Betätigung des Zurück-Buttons mit dem Titel *Navigation* in der NavigationBar des MasterViewControllers werden wieder die verfügbaren PDF Dateien im MasterViewController angezeigt. Auf den Inhalt des DetailViewControllers hat dies keinen Einfluss, solange keine andere PDF Datei ausgewählt wird.

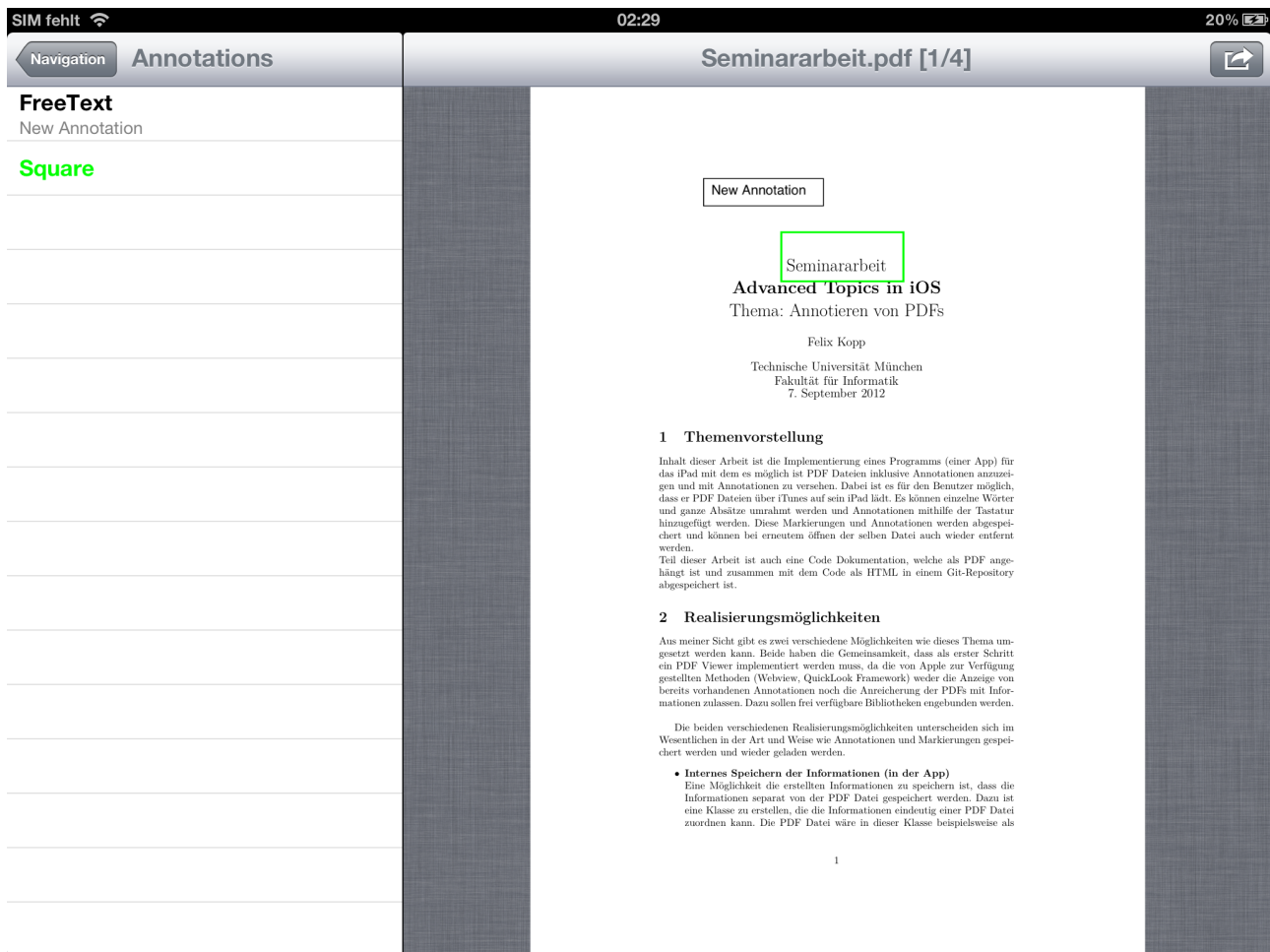


Abbildung 2: Benutzeroberfläche nach Auswahl einer Datei

Bei Betätigung des Action-Buttons in der NavigationBar des DetailViewController rechts, öffnet sich ein Popover für neue Annotationen. Dieses Popover wird in Abbildung 3 dargestellt. Hier getätigte Einstellungen werden bei der Erstellung neuer Annotationen verwendet. Es werden nur FreeText und Square Annotationen unterstützt.

Um eine neue Annotation zu erstellen muss eine LongPress Gesture an der Stelle ausgeführt werden, wo die Annotation erstellt werden soll. Bei FreeText Annotationen erscheint dann ein TextView, in den man seinen Text eingeben kann. FreeText Annotationen werden gespeichert, sobald die Tastatur eingefahren wird. Bei Square Annotationen muss man die LongPress Gesture halten und das gewünschte Rechteck formen. Square Annotationen werden gespeichert sobald die LongPress Gesture beendet wird.

Bei Verwendung der App im Portrait Modus ist die Bedienung die Selbe, nur dass der MasterViewController in einem Popover dargestellt wird.

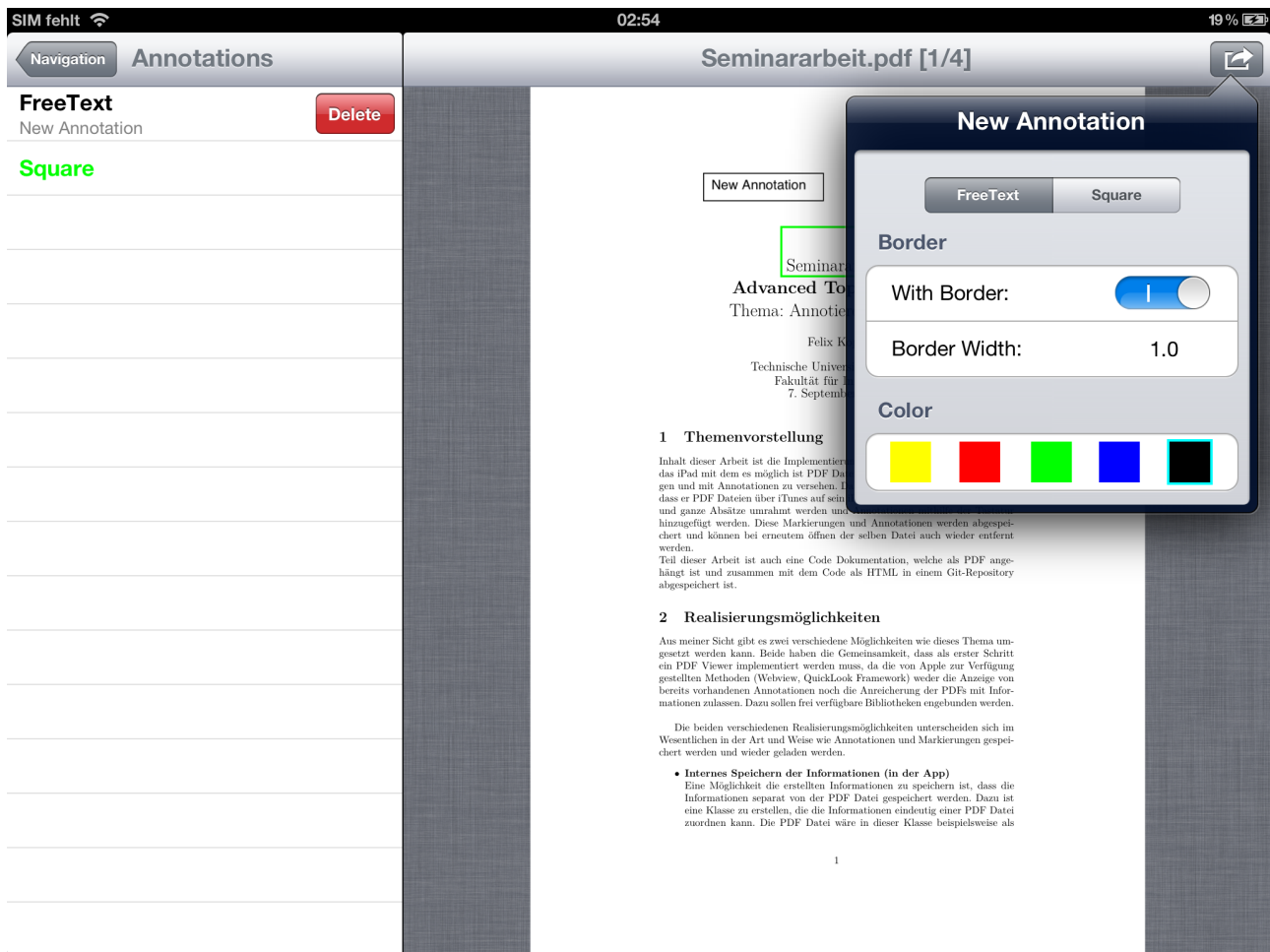


Abbildung 3: Benutzeroberfläche zeigt das Löschen einer Annotation und New Annotation Popover

4.2 Implementierung

Der Sourcecode der App beinhaltet eine Codedokumentation im HTML Format und im PDF Format. Darin sind alle Klassen, Methoden und Attribute dokumentiert. Um die Implementierung zu verstehen sollte diese Dokumentation betrachtet werden. Hier wird nur die generelle Funktionsweise behandelt.

- **Anzeigen der PDFs**

Zur Anzeige der PDF Dateien wird das QuartzCore Framework verwendet. Damit ist es möglich mithilfe der Methode *CGContextDrawPDFPage()* eine PDF Seite in einen gewünschten Context zu zeichnen. Allerdings werden Annotationen hierbei nicht berücksichtigt. Um Annotationen anzuzeigen werden UIViews verwendet. Je Annotation wird dieser UIView so angepasst, dass er den Parametern der Annotation entspricht. Für die Anzeige unterstützte Annotationstypen sind: Highlight, Square

und FreeText. Bei Highlight und Square Annotationen ist kein Text im UITextView.

Bei **Highlight Annotationen** wird die Hintergrundfarbe des UITextViews auf die Farbe der Highlightfarbe und der alpha Wert auf 0.5 gesetzt um den Text sichtbar zu halten. Der frame des UITextViews entspricht dem Frame der Annotation.

Bei **Square Annotationen** wird die Hintergrundfarbe des UITextViews auf transparent gesetzt und ein Rahmen in der Squarefarbe um den UITextView gezeichnet. Der frame des UITextViews entspricht dem Frame der Annotation.

Bei **FreeText Annotationen** wird die Hintergrundfarbe des UITextViews auf transparent gesetzt und der Text des UITextViews entspricht dem der Annotation. Wenn in der Annotation ein Rahmen definiert ist wird ein Rahmen um den UITextView gezeichnet.

Um die Annotationen nun anzuzeigen wird die Methode `[textView.layer renderInContext:context]` verwendet, wobei der Context der Selbe ist, auf den `CGContextDrawPDFPage()` angewendet worden ist.

- **Manipulation und Speichern der PDFs**

Zur Manipulation und zum Speichern der PDFs wird die PoDoFo Library verwendet. Damit werden Annotationen direkt in die PDF Datei geschrieben und auch direkt aus ihr gelöscht. Weiter wird damit die PDF Datei nach Anreicherung mit einer Annotation oder nach dem löschen einer Annotation wieder in das Shared Documents Verzeichnis gespeichert. Alle Methoden, die dazu gebraucht werden sind in der Klasse APDFManager als Klassenmethoden implementiert.

Mit PoDoFo ist es nicht möglich geöffnete Dateien in die selbe Datei zurück zu speichern. Deshalb wird beim **Speichern** die PDF Datei in eine temporäre Datei mit Namen tmp.apdf gespeichert. Anschließend wird mit dem NSFileManager die Datei tmp.apdf zu der ursprünglichen Datei gemacht.

5 Verwendete Frameworks/Libraries

In diesem Abschnitt wird kurz auf das verwendete QuartzCore Framework und die PoDoFo Library eingegangen.

- **QuartzCore Framework**

Das QuartzCore Framework bietet Funktionen um Informationen aus PDF Dateien auszulesen und um PDF Seiten anzuzeigen. Es ist damit auch möglich Annotationen auszulesen. In dieser App werden vorhandene Annotationen auch mithilfe des QuartzCore Frameworks ausgelesen und dann mithilfe von UITextViews angezeigt. Allerdings bietet dieses Framework keine Funktionen an um Annotationen anzuzeigen oder um PDFs zu manipulieren.

- **PoDoFo Library**

Die PoDoFo Library bietet eine Vielzahl an Funktionen um PDF Dateien

zu manipulieren und auch um sie zu erstellen. Wer mehr zu den Funktionen wissen will, sei auf die Dokumentation der Library verwiesen[2]. In dieser App wird diese Library dazu verwendet um PDF Annotationen zu erstellen und manipulierte (mit Annotationen versehene) PDF Dateien zu speichern.

6 Probleme bei der Implementierung

In diesem Abschnitt werden Probleme genannt, die bei der Implementierung der App aufgetreten sind.

6.1 Darstellung der Annotationen

Bei der Darstellung der Annotationen besteht das Problem darin, dass Annotationen separat von der dazugehörigen PDF Seite gerendert werden müssen und das Koordinatensystem bei PDFs ihren Ursprung links unten hat. Dadurch muss die Berechnung der Position einer Annotation möglichst exakt sein um eine Verschiebung zur PDF Seite zu vermeiden.

6.2 Einbindung der PoDoFo Library

Bei der Einbindung der PoDoFo Library besteht das Problem darin, dass iOS als Plattform nicht offiziell unterstützt wird. Das liegt unter anderem daran, weil PoDoFo auf Libraries zugreift, wie beispielsweise FontConfig, die in iOS nicht integriert sind. Somit müssen, um ein Einbinden der PoDoFo Library zu ermöglichen, auch andere Libraries eingebunden werden. Das größte Problem bereitete dabei die bereits erwähnte FontConfig Library. In der FontConfig Library sind Pfade eingetragen, die auf Dateien verweisen, die beispielsweise auf dem Mac liegen. Diese Pfade lassen sich allerdings nicht verändern, womit es bei einem Aufruf der Library zu einem Absturz der App kommt.

Letztendlich wurde das Problem behoben, indem Environment Variablen für *FONTCONFIG_FILE* und *FONTCONFIG_PATH* beim Start der App erstellt werden.

7 Funktionalitäten

Die Applikation, die mit dieser Arbeit implementiert wurde beinhaltet folgende Funktionalitäten:

- **Anzeigen von PDF Dateien**

Wie schon erwähnt ist es nicht möglich die in iOS integrierten PDF Viewer zu verwenden, da das Anzeigen von Annotationen mit diesen Viewern nicht möglich ist. In dieser Arbeit ist deshalb ein PDF Viewer der das Anzeigen von Annotationen ermöglicht, implementiert worden und das Annotieren mit FreeText Annotationen und das Markieren mit Square Annotationen ermöglicht.

- **Hinzufügen von Annotationen und Markierungen**

Eine weitere Funktionalität die integriert worden ist, ist das hinzufügen

von FreeText Annotationen und Square Markierungen. So ist es dem User möglich beliebig Wörter oder Absätze zu markieren und Annotationen an beliebigen Stellen hinzuzufügen. Annotationen können mithilfe der Tastatur erstellt werden.

- **Speichern der Informationen in der PDF Datei**

Ebenso ist die Funktionalität implementiert worden, dass die erstellten Informationen gespeichert werden. Dazu werden alle vom Benutzer erstellten Annotationen und Markierungen direkt in die PDF Datei geschrieben werden.

- **Änderbarkeit der Informationen**

Weiter ist die Funktionalität, dass bereits erstellte Informationen wieder geladen und auch verändert werden können enthalten. Eine Veränderung ist insofern möglich, dass die Informationen gelöscht werden können.

- **Export der annotierten PDFs**

Eine weitere Funktionalität die implementiert ist, ist dass die annotierten PDF Dateien aus dem Shared Documents Ordner via iTunes exportiert werden können. Dadurch ist es für den Benutzer möglich seine annotierten PDF Dateien auch an einem Desktop Rechner zu betrachten.

Literatur

- [1] http://www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/pdf_reference_1-7.pdf, 6th September 2012.
- [2] <http://podofo.sourceforge.net/doc/html/>, 6th September 2012.