

**Projektarbeit (Informatik)**

Sichere eTest auf eBook Readern

|  |  |  |
| --- | --- | --- |
| **Autoren** |  | Simon Lukes  Daniel Jampen |
| **Hauptbetreuung** |  | Karl Rege |
| **Datum** |  | 04.12.2014 |



Zusammenfassung

Abstract… english

Inhaltsverzeichnis

[1 Einleitung 8](#_Toc405393119)

[1.1 Ausgangslage 8](#_Toc405393120)

[1.2 Problemstellung 8](#_Toc405393121)

[1.3 Zielsetzung / Aufgabenstellung / Anforderungen 8](#_Toc405393122)

[1.3.1 Muss Funktionen 8](#_Toc405393123)

[1.3.2 Wunsch Funktionen 8](#_Toc405393124)

[2 Grundlagen 9](#_Toc405393125)

[2.1 Exportformat 9](#_Toc405393126)

[2.1.1 PDF 9](#_Toc405393127)

[2.1.2 HTML 9](#_Toc405393128)

[2.2 Kryptographie 9](#_Toc405393129)

[2.2.1 Verschlüsselung 9](#_Toc405393130)

[2.2.2 Passwort Hashing 10](#_Toc405393131)

[2.2.3 INI 11](#_Toc405393132)

[2.2.4 XML 11](#_Toc405393133)

[2.3 Import 11](#_Toc405393134)

[2.3.1 XML 11](#_Toc405393135)

[2.3.2 Open Office 11](#_Toc405393136)

[2.3.3 Word 11](#_Toc405393137)

[2.3.4 HTML 11](#_Toc405393138)

[2.4 Digitale Manipulationsmöglichkeiten 11](#_Toc405393139)

[2.4.1 Zeit 11](#_Toc405393140)

[2.4.2 Internet Zugriff 12](#_Toc405393141)

[2.4.3 Prüfungsinterne Kommunikation 12](#_Toc405393142)

[3 Analyse 12](#_Toc405393143)

[3.1 Anwendungsfälle 12](#_Toc405393144)

[3.1.1 Übersicht 13](#_Toc405393145)

[3.1.2 Akteure 13](#_Toc405393146)

[3.1.3 AF01… 13](#_Toc405393147)

[3.2 Nichtfunktionale Anforderungen 13](#_Toc405393148)

[3.2.1 NFA01.. 13](#_Toc405393149)

[4 Konzept 13](#_Toc405393150)

[4.1 High-Level 13](#_Toc405393151)

[4.1.1 Domänenmodell 13](#_Toc405393152)

[4.2 Export 13](#_Toc405393153)

[4.2.1 Vergleichsmatrix 13](#_Toc405393154)

[4.2.2 Entscheid 13](#_Toc405393155)

[4.3 Kryptographie 14](#_Toc405393156)

[4.3.1 Verschlüsselung 14](#_Toc405393157)

[4.3.2 Passwort hashing 16](#_Toc405393158)

[4.4 Konfigurationsdateien 16](#_Toc405393159)

[4.4.1 Vergleichsmatrix 16](#_Toc405393160)

[4.4.2 Entscheid 16](#_Toc405393161)

[4.5 Import 16](#_Toc405393162)

[4.5.1 Vergleichsmatrix 16](#_Toc405393163)

[4.5.2 Entscheid 16](#_Toc405393164)

[4.6 Digitale Manipulationsmöglichkeiten 16](#_Toc405393165)

[4.6.1 Zeit 16](#_Toc405393166)

[4.6.2 Internet Zugriff 16](#_Toc405393167)

[4.6.3 Prüfungsinterne Kommunikation 16](#_Toc405393168)

[4.7 UI-Design 16](#_Toc405393169)

[4.8 JavaScript Library 16](#_Toc405393170)

[4.8.1 SecureExam 16](#_Toc405393171)

[4.8.2 CryptoJS 16](#_Toc405393172)

[4.8.3 FileSaver.js 16](#_Toc405393173)

[5 Umsetzung 16](#_Toc405393174)

[5.1 Export 17](#_Toc405393175)

[5.2 Kryptographie 17](#_Toc405393176)

[5.2.1 Verschlüsselung 17](#_Toc405393177)

[5.2.2 Passwort hashing 17](#_Toc405393178)

[5.3 Konfigurationsdateien 17](#_Toc405393179)

[5.3.1 Settings.xml 17](#_Toc405393180)

[5.3.2 SecureExam.xml 17](#_Toc405393181)

[5.4 Import 17](#_Toc405393182)

[5.4.1 XML 17](#_Toc405393183)

[5.4.2 Open Office 17](#_Toc405393184)

[5.5 Digitale Manipulationsmöglichkeiten 17](#_Toc405393185)

[5.5.1 Zeit 17](#_Toc405393186)

[5.5.2 Internet Zugriff 17](#_Toc405393187)

[5.5.3 Prüfungsinterne Kommunikation 17](#_Toc405393188)

[5.6 UI-Design 17](#_Toc405393189)

[5.7 JavaScript Library 17](#_Toc405393190)

[6 Testing 17](#_Toc405393191)

[6.1 Konzept 17](#_Toc405393192)

[6.2 C# Code 17](#_Toc405393193)

[6.3 HTML / JS Funktionalität 17](#_Toc405393194)

[7 Resultate 17](#_Toc405393195)

[8 Diskussion und Ausblick 17](#_Toc405393196)

[9 Verzeichnisse 18](#_Toc405393197)

[9.1 Literaturverzeichnis 18](#_Toc405393198)

[9.2 Glossar 18](#_Toc405393199)

[9.3 Abbildungsverzeichnis 18](#_Toc405393200)

[9.4 Tabellenverzeichnis 18](#_Toc405393201)

[9.5 Abkürzungsverzeichnis 18](#_Toc405393202)

[10 Anhang 18](#_Toc405393203)

[10.1 Projektmanagement 18](#_Toc405393204)

[10.1.1 Zeitplan 18](#_Toc405393205)

[10.1.2 Sitzungsprotokolle 18](#_Toc405393206)

[10.2 C# Klassendiagramm 18](#_Toc405393207)

[10.3 Bedienungsanleitung 18](#_Toc405393208)

[10.3.1 Konsolenapplikation 18](#_Toc405393209)

[10.3.2 OpenOffice Import 18](#_Toc405393210)

[10.3.3 Generierte Prüfung 18](#_Toc405393211)

[10.4 Weiteres 18](#_Toc405393212)

[10.4.1 CD mit dem vollständigen Bericht als pdf-File und dem SourceCode 18](#_Toc405393213)

# Einleitung

Momentan werden wichtige Prüfungen fast ausschliesslich in Papierform durchgeführt. Dies trotz den immer besser werdenden technischen Möglichkeiten, Prüfungen in digitaler Form durchzuführen zu können.

Nicht nur wäre eine Umstellung auf ein digitales Format eine grosse Erleichterung für die Entlastung der Korrekturarbeit der Professoren, sondern auch für die Studenten. Zum Beispiel könnten Programmierprüfungen in gewohntem Umfeld (über die Tastatur geschriebener Sourcecode) abgehalten werden, oder auch Antworten könnten endlich sauber überarbeitet werden. Diverse dieser Verbesserungsmöglichkeiten sind sehr einfach mit digitalen Prüfungen umsetzbar, doch leider gibt es auch ein paar Schwachpunkte.

Dadurch, dass man digital arbeitet, müssen auch die Manipulationsschutzmechanismen überarbeitet werden. Mit den vielen, heute verfügbaren, technischen Hilfsmitteln, reicht es schon lange nicht mehr aus, einfach die Prüfungsteilnehmer nach Spickzetteln zu durchsuchen.

## Ausgangslage

* Nennt bestehende Arbeiten/Literatur zum Thema  Literaturrecherche
* Stand der Technik: Bisherige Lösungen des Problems und deren Grenzen

## Problemstellung

* Nicht sicher => manipulationen
* Prüfungsfile kommt abhanden
* Infrastruktur unzuverlässig
* Kommunikationsfähigkeiten
* …

## Zielsetzung / Aufgabenstellung / Anforderungen

### Muss Funktionen

### Wunsch Funktionen

|  |  |
| --- | --- |
| Suchen einer Erinnerung | Durchsuchen der persönlichen Erinnerungen. Kriterien sind Stichworte, Daten, Status und weitere. |
| Filtern der Erinnerungen | Filtern der persönlichen Erinnerungen. Kriterien sind Stichworte, Daten, Status und weitere. |
| Administrationswerkzeug | Mit diesem Werkzeug lassen sich Benutzerkonten verwalten und Änderungen an der Website vornehmen. |

* Formuliert das Ziel der Arbeit
* (Pflichtenheft, Spezifikation)
* (Spezifiziert die Anforderungen an das Resultat der Arbeit)
* (Übersicht über die Arbeit: stellt die folgenden Teile der Arbeit kurz vor)
* (Angaben zum Zielpublikum: nennt das für die Arbeit vorausgesetzte Wissen)

# Grundlagen

## Exportformat

### PDF

### HTML

## Kryptographie

### Verschlüsselung

Die Verschlüsselung eines Datensatzes dient generell dazu, dessen Inhalt vor unberechtigten Personen zu verbergen (Confidentiality). Die zwei am weitest verbreitetsten Methoden sind die Secret Key Verschlüsselung und die Public Key Verschlüsselung. Anschliessend wird jeweils der aktuelle Standard beschrieben.

#### AES

AES wurde im Jahre 2001 offiziell publiziert und ist der momentane Standard bei der „Secret Key“ Verschlüsselung. AES verschlüsselt den gegebenen Input nicht als Ganzes, sondern unterteilt diesen in kleine Blöcke (128 bis 258 Byte), welche letztendlich Stück für Stück verschlüsselt werden.



Abbildung : AES Block Verschlüsselung

Für die Verschlüsselung der Blöcke stehen zwei verschiedene Modi zur Verfügung:

|  |  |
| --- | --- |
| ECB    Abbildung 2: ECB Verschlüsselungsmodus  Jeder Block wird unabhängig betrachtet und mit dem gleichen Schlüssel verschlüsselt. | CBC    Abbildung 3: CBC Verschlüsselungsmodus  CBC Verschlüsselt dem ersten Block mit einem Initialisierungsvektor (IV) und die folgenden Blöcke mit dem Ciphertext des jeweils vorherigen Blockes. |

**Quelle: ISI Vorlesung**

CBC bietet im Vergleich zu ECB eine erhöhte Sicherheit gegen Manipulationen am Ciphertext. Dadurch, dass die Blöcke abhängig voneinander Verschlüsselt werden, würde beim Austauschen eines Blockciphers der jeweils vorherige beschädigt.

#### RSA

RSA ist der Standard bei der Public / Private Key Verschlüsselung. Dies bedeutet, dass für jeden Teilnehmer ein Private - sowie Public Key generiert werden muss. Mittels Signierung der Zertifikate von einer global anerkannten „Certificate Authority“ kann zusätzlich die Echtheit eines Zertifikates ermittelt werden.



Abbildung : RSA Verschlüsselung

User 2 Verschlüsselt seine Nachricht an User 1 mittels dem frei Verfügbaren Public Key von User 1 und überträgt anschliessend die verschlüsselte Nachricht an den Empfänger. User 1 kann nun die Nachricht mit seinem Private Key entschlüsseln und die Originalnachricht lesen.

### Passwort Hashing

Kryptographische Hashfunktionen berechnen aus einem variabel langen Input einen fixen Output (den Hash). Dieser muss folgende Eigenschaften aufweisen:

* Pseudo zufällig: Wenn ein Bit des Inputs verändert wird, müssen sich ca. 50% der Outputbits auch ändern.
* One Way: mit einem gegebenen Hash muss es praktisch unmöglich sein, die Ursprungs-nachricht zu ermitteln.
* Collision free: Es muss praktisch unmöglich sein, zwei Nachrichten zu finden, welche den selben Hash produzieren.

Typische Hash-Längen (Output Länge) sind je nach gewähltem Algorithmus zwischen 128 und 512 Bits lang.

Beim Passwort Hashing wird oft ein Hash-Chaining angewendet. Dies bedeutet, dass die Hash-funktion mehrmals nacheinander ausgeführt wird, um die Berechnungszeit des Hashs zu erhöhen. Ein Brute-Force Angriff auf den Hash dauert, je mehr Hash-Iterationen gemacht werden, entsprechend länger.

#### SHA3

Die momentan neuste Version der SHA Algorithmen, welche vom US-Amerikanischen National Institute of Standards and Technology (NIST), veröffentlicht werden. Die verfügbaren Hashlängen sind zwischen 224 und 512 Bit, was eine Sicherheit gegen Kollisionsangriffe von 256 Bit bietet (wegen des Geburstagsparadoxum).

#### PBKDF2

PBKDF2 steht für Password Based Key Derivation Function und wurde entwickelt, um von einem Passwort einen Key abzuleiten, welcher in einem symmetrischen Verschlüsselungsverfahren ver-wendet werden kann.

PBKDF2 wendet auf den Input einen Hashalgorithmus mit Salting und anschliessendem Chaining an. Konfigurationsdateien

### INI

### XML

## Import

### XML

### Open Office

### Word

### HTML

## Digitale Manipulationsmöglichkeiten

Sobald Prüfungen in digitaler Form abgelegt werden können, gibt es immer Bedenken betreffend Manipulationsmöglichkeiten während bzw. an der Prüfung. Nachfolgend sind die möglichen Manipulationen aufgelistet, die bei Prüfungen auf eBook Readern auftreten könnten.

### Korrekte Antwort auslesen

Für die Korrektur der Antworten sind bereits die korrekten Antworten irgendwo in der Prüfung gespeichert, wenn auch nicht sichtbar für den Prüfungsteilnehmer. Es wäre also mit genügend Fachwissen möglich, diese auszulesen und somit die Prüfung vollständig korrekt zu beantworten.

### Zeit

Die Restzeit der Prüfung wird digital berechnet. Falls es nun möglich wäre die Zeit zu manipulieren, könnte sich ein Teilnehmer eine grössere Zeitspanne für die Prüfung ermöglichen. Um dies zu realisieren wären folgende zwei Szenarien möglich:

#### Manipulation der geräteinternen Uhr

Angenommen die Zeitspanne bis zum Ende der Prüfung berechnet sich mittels folgender Formel:

Dann wäre es möglich, nach dem Starten der Prüfung, die interne Uhr um eine gewisse Zeitspanne zurückzusetzen um dem Kandidat mehr Zeit für die Prüfung zu generieren.

#### Verlangsamung der Zeit

Um die Zeitmanipulation besser zu tarnen, könnte in regelmässigen Abständen, die Zeit um ein paar Millisekunden zurückgesetzt werden. Angenommen, man würde alle 10ms die Zeitrechnung um 5ms zurücksetzen, dann würde eine Sekunde der Systemuhr plötzlich zwei reale Sekunden dauern. Der Kandidat hätte also die doppelte Zeitspanne zum Lösen der Prüfung zur Verfügung.

### Internet Zugriff

Dadurch dass fast alle Geräte heutzutage mit Wireless Modulen ausgestattet sind, wäre es möglich, während der Prüfung im Internet nach Lösungen der Prüfungsfragen zu suchen.

### Prüfungsinterne Kommunikation

Sobald technische Geräte mit Kommunikationsfunktionen ausgestattet sind, können diese auch missbraucht werden. Probanden hätten somit die Möglichkeit, während der Prüfung untereinander zu kommunizieren. Aktuelle eBook Reader könnten folgende Kommunikationsmodule verbaut haben:

#### Wireless LAN

Auch wenn der Prüfungsraum gegen jegliche Wirelessstrahlung von aussen abgeschirmt wäre, wäre es trotzdem möglich, dass intern Ad-Hoc Netzwerke erstellt würden. Ein Teilnehmer könnte ein solches erstellen und die anderen Probanden könnten diesem beitreten und darüber kommunizieren.

#### Bluetooth Modul

Mittels Bluetooth sind Datenverbindungen zwischen zwei Endgeräten möglich. Somit könnten Kandidaten durch einschalten von Bluetooth miteinander Daten austauschen / miteinander kommunizieren.

## eBook Reader

Als eBook Reader werden technische Geräte bezeichnet, welche primär zum Lesen von Büchern erstellt und optimiert wurden. Für die Eigenschaften der Geräte bedeutet dies, dass der Fokus auf langer Akkulaufzeit und angenehmem lesen von Texten auf dem Bildschirm liegt.

Um eine lange Akkulaufzeit zu erreichen, werden oft eher langsame elektronische Komponenten verbaut und es werden eingeschränkte, speziell angepasste Betriebssysteme verwendet. Es wird nachfolgend generell davon ausgegangen, dass eBook Reader kein voll funktionsfähiges Android OS besitzen. Solchen Geräte unterscheiden sich kaum mehr von Tablets und würden als solche angesehen.

### Einschränken

#### Betriebssystem

Die Betriebssysteme von eBook Readern sind meist vereinfachte Versionen von Android (z.B. Tolino OS) oder vom Hersteller selber entwickelte Linux Distributionen(z.B. nicht Android basierte Amazon Kindle Geräte). Die Idee dahinter ist, ein einfaches, Ressourcen schonendes System zu haben, welches gerade nur die Funktionen bietet, die zum Lesen von eBooks benötigt werden.

#### CPU / GPU Geschwindigkeit

EBook Reader benötigen für den normalbetrieb sehr wenig Rechengeschwindigkeit. Die CPU und GPU werden so gewählt, dass in akzeptabler Zeit eine Buchseite dargestellt werden kann und dass der Endbenutzer eine möglichst lange Akkulaufzeit hat. Es steht somit deutlich weniger Rechenleistung zur Verfügung als bei aktuellen Tablets.

#### Displaytechnik

Bei eBook Readern werden im Gegensatz zu den Tablets sogenannte E-Ink Displays verwendet. Diese Displays brauchen nur Strom, wenn das Bild geändert wird. Eine Änderung wäre zum Beispiel das wechseln von einer Buchseite auf eine andere. Ein statisches Bild bleibt ohne zusätzlich benötigte Energie über einen, je nach Display variierenden, Zeitraum erhalten. Der Fokus der Technik liegt ganz klar bei der Energieeffizienz und der guten Lesbarkeit bei möglichst vielen Lichtverhältnissen.

# Analyse

## Anwendungsfälle

Sicht? Anwender!

### Übersicht

### Akteure

### AF01 Prüfung erstellen

### AF02 Prüfung absolvieren

### AF02 Abgelegte Prüfung auslesen

## Nichtfunktionale Anforderungen

### NFA01.. ???

# Konzept

## Systemüberblick

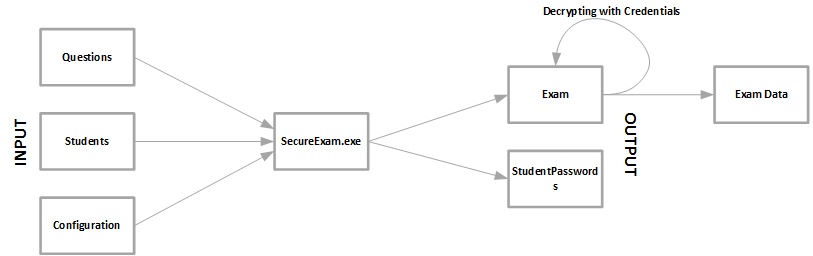
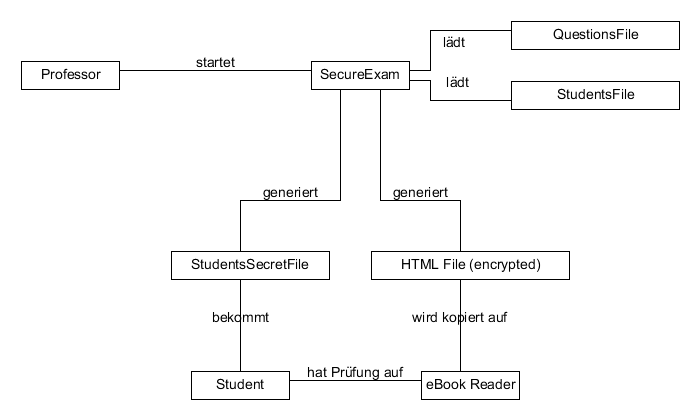


Abbildung : Systemüberblick

Das Diagramm bietet eine Übersicht über die Komponenten. Die Applikation SecureExam.exe generiert aus den drei Eingabedateien die Prüfung sowie eine Datei mit den Passwörtern für die Studenten. Beim Öffnen der Prüfungsdatei wird der Student aufgefordert, seine erhaltenen Daten (Vorname, Nachname, Immatrikulationsnummer, Zufallspasswort) einzugeben. Sobald diese korrekt eingegeben wurden, entschlüsselt sich die Prüfung selber und die Fragen werden angezeigt. Beim Abschluss der Prüfung werden die Antworten exportiert.

### Domänenmodell



**Braucht wohl Überarbeitung…**

## Export

### Vergleichsmatrix

### Entscheid

## Kryptographie

### Verschlüsselung

#### Evaluationsmatrix



Tabelle : Vergleich AES und RSA

#### Entscheid

Wie bei der Vergleichsmatrix (vgl. 4.3.1.1) ersichtlich, hat AES gegenüber RSA einen gewaltigen Vorteil in Bezug auf die „einfache Keyeingabe“. Bei AES kann man ein normales Passwort mit anschliessendem Hashing verwenden, bei RSA müsste ein komplettes Zertifikat korrekt abgetippt werden. Deshalb wird AES als Verschlüsselungsverfahren eingesetzt.

#### Konzept

Für eine bessere Übersicht wird der Ver- bzw. Entschlüsselungsvorgang getrennt beschrieben. Die Verschlüsselung wird automatisch mittels dem Prüfungsgenerator durchgeführt, die Entschlüsselung hingegen findet auf dem Client direkt beim Starten der Prüfung statt.

##### Verschlüsselung



Abbildung : Multi-User AES Verschlüsselung

Für jeden Student wird ein individuelles Passwort generiert. Dieses setzt sich aus folgenden Komponenten zusammen:

* Vorname
* Nachname
* Immatrikulationsnummer
* n zufällige Zeichen (Base64)
* 128 Bit Salt.

Diese werden anschliessend n-Mal mit gehasht (vgl. 4.3.2, Passwort hashing). Mit dem resultierenden Hash (256 Bit) und einem zufälligen IV (Initialisierungsvektor) wird nun der Masterkey mittels AES verschlüsselt. Der Masterkey setzt sich aus einer zufälligen, 256 Bit langen Zeichenkette zusammen. Dieser wird dazu verwendet, um die Prüfungsdaten (Fragen, Prüfungszeitraum und Prüfungsdauer) zu verschlüsseln.

##### Entschlüsselung



Abbildung : Multi-User AES Entschlüsselung

Jeder Student wird beim Starten der Prüfung seine Zugangsdaten eingeben müssen. Diese bestehen aus:

* Vorname
* Nachname
* Immatrikulationsnummer
* n zufällige Zeichen

Mit diesen Informationen plus dem zugehörigen Salt, wird wie bei der Verschlüsselung, ein Hashing durchgeführt. Das resultierende „Student Password“ dient als Passwort für die Entschlüsselung des Masterkeys. Mittels diesem und dem zugehörigen IV, werden dann die eigentlichen Daten entschlüsselt.

### Passwort hashing

#### Evaluationsmatrix

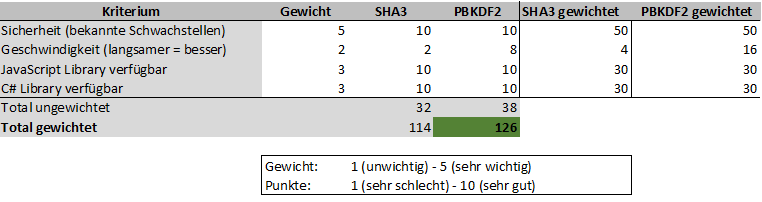


Tabelle : Vergleich SHA3 und PBKDF2

#### Entscheid

Die beiden Algorithmen sind praktisch gleichauf. PBKDF2 ist ein Algorithmus, welcher absichtlich langsam, für das generieren eines Keys aus einem Passwort entwickelt wurde. Trotzdem wird bei SecureExam SHA3 als Hash-Algorithmus eingesetzt. Der Entscheid wurde so getroffen, um eine bessere Anpassung an langsame Hardware (eBook Reader) zu ermöglichen.

#### Hashing Konzept

Um die Zeit, welche zum Hashen benötigt wird, optimal an die eBook Reader Hardware anzupassen, ist die Anzahl Iterationen beim Chaining Vorgang via Konfigurationsparameter anpassbar.



Abbildung : Hashing Ablauf

Die Studenteninformationen werden zusammen mit n zufälligen Zeichen (Passwort) und einem kryptografisch sicheren, zufälligen Salt gehasht. Der resultierende Hash wird noch (n-1) Mal mittels SHA3 gehasht und der letzte Hashwert wird als Resultat ausgegeben und als Key verwendet.

## Konfigurationsdateien

### Vergleichsmatrix

### Entscheid

## Import

### Vergleichsmatrix

### Entscheid

## Digitale Manipulationsmöglichkeiten

### Zeit

### Internet Zugriff

### Prüfungsinterne Kommunikation

## UI-Design

Papier virtualisieren etc.

Handskizze

## JavaScript Library

### SecureExam

### CryptoJS

### FileSaver.js

# Umsetzung

Mit Code Snipplets

## Export

## Kryptographie

### Verschlüsselung

### Passwort hashing

## Konfigurationsdateien

### Settings.xml

### SecureExam.xml

## Import

### XML

### Open Office

XSLT über eigenen Parser bla bli bla bla

## Digitale Manipulationsmöglichkeiten

### Zeit

### Internet Zugriff

### Prüfungsinterne Kommunikation

## UI-Design

## JavaScript Library

# Testing

## Konzept

## C# Code

## HTML / JS Funktionalität

# Resultate

Lauffähig auf eBook-Reader und Tablet => bilder bla bla

# Diskussion und Ausblick

* Bespricht die erzielten Ergebnisse bezüglich ihrer Erwartbarkeit, Aussagekraft und Relevanz
* Interpretation und Validierung der Resultate
* Rückblick auf Aufgabenstellung, erreicht bzw. nicht erreicht
* Legt dar, wie an die Resultate (konkret vom Industriepartner oder weiteren Forschungsarbeiten; allgemein) angeschlossen werden kann; legt dar, welche Chancen die Resultate bieten
* Ausblick

# Verzeichnisse

## Literaturverzeichnis

## Glossar

## Abbildungsverzeichnis

## Tabellenverzeichnis

## Abkürzungsverzeichnis

# Anhang

## Projektmanagement

### Zeitplan

### Sitzungsprotokolle

## C# Klassendiagramm

[Bild]

Modularität bla bla bla

## Bedienungsanleitung

### Konsolenapplikation

### OpenOffice Import

### Generierte Prüfung

## Weiteres

### CD mit dem vollständigen Bericht als pdf-File und dem SourceCode