

**Projektarbeit (Informatik)**

Sichere eTest auf eBook Readern

|  |  |  |
| --- | --- | --- |
| **Autoren** |  | Simon Lukes  Daniel Jampen |
| **Hauptbetreuung** |  | Karl Rege |
| **Datum** |  | 03.12.2014 |

Ehrenwörtliche Erklärung..

Zusammenfassung

Abstract…

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 11](#_Toc405393141)

[2.4.3 Prüfungsinterne Kommunikation 11](#_Toc405393142)

[3 Analyse 11](#_Toc405393143)

[3.1 Anwendungsfälle 11](#_Toc405393144)

[3.1.1 Übersicht 12](#_Toc405393145)

[3.1.2 Akteure 12](#_Toc405393146)

[3.1.3 AF01… 12](#_Toc405393147)

[3.2 Nichtfunktionale Anforderungen 12](#_Toc405393148)

[3.2.1 NFA01.. 12](#_Toc405393149)

[4 Konzept 12](#_Toc405393150)

[4.1 High-Level 12](#_Toc405393151)

[4.1.1 Domänenmodell 12](#_Toc405393152)

[4.2 Export 12](#_Toc405393153)

[4.2.1 Vergleichsmatrix 12](#_Toc405393154)

[4.2.2 Entscheid 12](#_Toc405393155)

[4.3 Kryptographie 13](#_Toc405393156)

[4.3.1 Verschlüsselung 13](#_Toc405393157)

[4.3.2 Passwort hashing 15](#_Toc405393158)

[4.4 Konfigurationsdateien 15](#_Toc405393159)

[4.4.1 Vergleichsmatrix 15](#_Toc405393160)

[4.4.2 Entscheid 15](#_Toc405393161)

[4.5 Import 15](#_Toc405393162)

[4.5.1 Vergleichsmatrix 15](#_Toc405393163)

[4.5.2 Entscheid 15](#_Toc405393164)

[4.6 Digitale Manipulationsmöglichkeiten 15](#_Toc405393165)

[4.6.1 Zeit 15](#_Toc405393166)

[4.6.2 Internet Zugriff 15](#_Toc405393167)

[4.6.3 Prüfungsinterne Kommunikation 15](#_Toc405393168)

[4.7 UI-Design 15](#_Toc405393169)

[4.8 JavaScript Library 15](#_Toc405393170)

[4.8.1 SecureExam 15](#_Toc405393171)

[4.8.2 CryptoJS 15](#_Toc405393172)

[4.8.3 FileSaver.js 15](#_Toc405393173)

[5 Umsetzung 15](#_Toc405393174)

[5.1 Export 16](#_Toc405393175)

[5.2 Kryptographie 16](#_Toc405393176)

[5.2.1 Verschlüsselung 16](#_Toc405393177)

[5.2.2 Passwort hashing 16](#_Toc405393178)

[5.3 Konfigurationsdateien 16](#_Toc405393179)

[5.3.1 Settings.xml 16](#_Toc405393180)

[5.3.2 SecureExam.xml 16](#_Toc405393181)

[5.4 Import 16](#_Toc405393182)

[5.4.1 XML 16](#_Toc405393183)

[5.4.2 Open Office 16](#_Toc405393184)

[5.5 Digitale Manipulationsmöglichkeiten 16](#_Toc405393185)

[5.5.1 Zeit 16](#_Toc405393186)

[5.5.2 Internet Zugriff 16](#_Toc405393187)

[5.5.3 Prüfungsinterne Kommunikation 16](#_Toc405393188)

[5.6 UI-Design 16](#_Toc405393189)

[5.7 JavaScript Library 16](#_Toc405393190)

[6 Testing 16](#_Toc405393191)

[6.1 Konzept 16](#_Toc405393192)

[6.2 C# Code 16](#_Toc405393193)

[6.3 HTML / JS Funktionalität 16](#_Toc405393194)

[7 Resultate 16](#_Toc405393195)

[8 Diskussion und Ausblick 16](#_Toc405393196)

[9 Verzeichnisse 17](#_Toc405393197)

[9.1 Literaturverzeichnis 17](#_Toc405393198)

[9.2 Glossar 17](#_Toc405393199)

[9.3 Abbildungsverzeichnis 17](#_Toc405393200)

[9.4 Tabellenverzeichnis 17](#_Toc405393201)

[9.5 Abkürzungsverzeichnis 17](#_Toc405393202)

[10 Anhang 17](#_Toc405393203)

[10.1 Projektmanagement 17](#_Toc405393204)

[10.1.1 Zeitplan 17](#_Toc405393205)

[10.1.2 Sitzungsprotokolle 17](#_Toc405393206)

[10.2 C# Klassendiagramm 17](#_Toc405393207)

[10.3 Bedienungsanleitung 17](#_Toc405393208)

[10.3.1 Konsolenapplikation 17](#_Toc405393209)

[10.3.2 OpenOffice Import 17](#_Toc405393210)

[10.3.3 Generierte Prüfung 17](#_Toc405393211)

[10.4 Weiteres 17](#_Toc405393212)

[10.4.1 CD mit dem vollständigen Bericht als pdf-File und dem SourceCode 17](#_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 vor-herigen 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 Geburts-tagsparadoxum).

#### 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

### Zeit

### Internet Zugriff

### Prüfungsinterne Kommunikation

# Analyse

## Anwendungsfälle

Sicht? Anwender!

### Übersicht

### Akteure

### AF01…

## Nichtfunktionale Anforderungen

### NFA01..

# Konzept

## High-Level

### Domänenmodell

## Export

### Vergleichsmatrix

### Entscheid

## Kryptographie

### Verschlüsselung

#### Vergleichsmatrix

#### Entscheid

#### 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

#### Vergleichsmatrix

#### Entscheid

#### Hashing Konzept

## 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 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