Post It

Modul 151

Zug

23.05.2022

Robin Ruzza & Julia Sorrentino

Inhaltsverzeichnis

[Informieren: 3](#_Toc104791695)

[Aufgabenstellung: 3](#_Toc104791696)

[Planen: 3](#_Toc104791697)

[Zieldefinition: 3](#_Toc104791698)

[Meilensteine: 4](#_Toc104791699)

[Entscheiden: 4](#_Toc104791700)

[Ausführen: 4](#_Toc104791701)

[4-Tier Architektur 4](#_Toc104791702)

[Datenbankanbindung/CRUD 4](#_Toc104791703)

[Datensicherheit 4](#_Toc104791704)

[Sessions/TRansationen/SSL/TLS 4](#_Toc104791705)

[Journal: 4](#_Toc104791706)

[Theorie 5](#_Toc104791707)

[Angriffsmöglichkeiten 5](#_Toc104791708)

[Ergriffene Massnahmen 5](#_Toc104791709)

[Kontrollieren: 5](#_Toc104791710)

[Bewerten: 5](#_Toc104791711)

# Informieren:

## Aufgabenstellung:

Für Ihre Applikation erstellen Sie ein GIT Repository. In der Dokumentation halten Sie fest welche Features/Funktionen Teile der Software von wem realisiert werden.

Diese Liste sollte mit Ihren regelmässigen Commits im Software Repo übereinstimmen.

Es wird erwartet das jeder Schüler/jede Schülerin mindestens 5 Gehaltvolle Commits leistet. Ein Commit zählt nur wenn er wesentliche Änderungen enthält.

Gibt es an einem Schultag keine Code-Änderungen dokumentieren Sie das in Ihrer Doku-Mutation. (Journal Ihrer Arbeiten)

Die Projektarbeit realisieren Sie in einer ausgewählten Technologie. Sie verwenden generell-le Techniken wie HTML, CSS und Javascript. Wenn Sie Frameworks einsetzen, begründen Sie die Wahl in der Dokumentation.

|  |  |
| --- | --- |
| Für die Bewertung der Arbeit werden unter anderem folgende Kriterien beachtet: Formale Anforderungen | Titel, Name, Thema, Datum, Seitenzahl usw. |
| Generelle Informationen | Spezifikation / Zieldefinition / Meilensteine |
| Architektur | Min. 4-Tier Architektur  Datenbankanbindung / CRUD  Datensicherheit wird berücksichtigt (Themen aus «Sicherheit in Web-Applikationen»)  Sessions / Transationen / SSL/TLS |
| Arbeitsjournal | Wer hat wann was gemacht? Es muss ersichtlich sein wer welche Arbeiten gemacht hat. |
| Datensicherheit | Ihre Anwendung ist gegen die gelernten Angriffe geschützt. In der Dokumentation gehen Sie auf die verschiedenen An-griffsmöglichkeiten sowie die ergriffenen Massnahmen ein. |

Sie dokumentieren alle Ihre Entscheide und Arbeiten in einem Dokument. Die Bewertung Ihrer Leistung findet Anhand des Dokuments statt.

Sie präsentieren Ihre Arbeit Ihren Kollegen. Bewertet werden folgende Kriterien

- Die Präsentation startet nach maximal zwei Minuten Vorbereitungszeit

- Die Präsentation dauert maximal 5 Minuten

- Einsatz von Präsentationsmitteln (z.B. Beamer, Tafel, Whiteboard usw.)

- Allgemeine Präsentationsmerkmale (z.B. Blick in Klasse, deutliches Sprechen usw.)

# Planen:

## Zieldefinition:

Unser Hauptziel ist es eine Vollständige und professionelle Dokumentation für unsere Software herzustellen. Dabei ist es wichtig, dass wir unser Journal bei jeglichen Änderungen/Erweiterungen/Updates in unserer Software festhalten und Dokumentieren. Wir müssen dabei das Abgabedatum sehr im Auge behalten, da wir es durch eine verspätete Abgabe einen Noten Abzug gibt.

Wir müssen auch jede Woche mindestens ein Commit machen, und darauf achten, dass jeder Schüler mindestens 5 Commits macht, somit müsste jeder einmal pro Woche ein Commit machen, damit alles aufgeht.

Ein weiterer wichtiger Teil dieses Projektes ist die Datensicherheit. Wir müssen unsere Software so gestalten und Schützen, dass Angriffe nicht bis nur leicht möglich sind. Die Massnahmen, die wir einführen werden, werden von Julia recherchiert und dokumentiert, und von Robin umgesetzt.

## Meilensteine:

1. Das Planen und Entscheiden von unserem Vorgehen
2. Das Grundgerüst der Software erstellen
3. Die Datenbank mit der Webseite verbinden
4. Die Datensicherheit zu recherchieren und in unserer Software anzuwenden
5. Das Testen unsere Software und das Abschliessen des Projektes
6. Die Dokumentation mit den letzten Informationen füllen

# Entscheiden:

Wir haben uns für die IPERKA Methode Entschieden und diese auch angepasst, damit wir ein Journal führen können und die Theorie, die wir benutzt haben erklären und begründen können.

# Realisieren:

## 4-Tier Architektur

## Datenbankanbindung/CRUD

## Datensicherheit

## Sessions/TRansationen/SSL/TLS

# Journal:

## Tag 1:

Als erstes haben wir uns zusammengesessen und entschieden welches Vorgehen wir benutzen wollten und wie haben uns für IPERKA entschieden. Danach haben wir die ersten kleineren Aufgaben aufgeteilt. Julia würde Grossteils die Dokumentation und die Theorie übernehmen während Robin sich um die Software kümmert. Als erstes fing Julia mit dem Grundgerüst der Dokumentation (IPERKA) an, und füllte dies mit den Anfangsinformationen (Das Informieren, das Planen und die Zieldefinition und der erste Tag des Journals)

Robin

## Tag 2:

Heute hat Julia mit der Dokumentation angefangen und hat sich mit dem zweiten Teil der Planung befasst, und hat somit die Meilensteine erstellt und diese mit Robin besprochen, und angepasst. Somit haben wir nun 6 Meilensteine. Nachdem der Planungsteil beendet war, und wir unser Ziel und Meilensteine wussten, hat Julia mit dem Testprotokoll angefangen, damit wir am Ende des Projektes die Ziele und Meilensteine testen können. Wir haben besprochen, ob wir nur ein Testprotokoll für die Software machen wollen, oder ob wir zwei Testprotokolle machen wollen. Eines für de Software inkl. der Datensicherheit und Datenbank und ein Testprotokoll für die allgemeine Organisation, Administration und Dokumentation.

Wir haben uns schlussendlich für ein einziges Testprotokoll entschieden, darin sollte die Software, die Datenbank und die Datensicherheit getestet.

Und somit, hat Julia nach der Besprechung mit dem Testprotokoll angefangen und hat zuerst das Grundgerüst des Testprotokolles gemacht. Dies hat sie danach auch zusammen mit der Dokumentation auf Github hochgeladen. Ihr ist dabei aufgefallen, dass sie für das Committen von Files noch hilfe brauchte, und somit schaute Sie auf Youtube ein kurzer Tutorial an.

# Theorie

## Angriffsmöglichkeiten

## Ergriffene Massnahmen

# Kontrollieren:

# Auswerten: