# IPA-Simulation

Raphael Pfeffinger

## Inhaltsverzeichnis

Inhalt

[IPA-Simulation 1](#_Toc169525157)

[Inhaltsverzeichnis 2](#_Toc169525158)

[1. Projektmanagement 3](#_Toc169525159)

[1.1 Projektmethodik 3](#_Toc169525160)

[1.2 Zeitplanung 4](#_Toc169525161)

[1.3 Testkonzept 4](#_Toc169525162)

[1.4 Arbeitsjournal 4](#_Toc169525163)

[2. Projektdokumentation 4](#_Toc169525164)

[2.1 Informieren (Analyse) 4](#_Toc169525165)

[2.2 Planen (Design) 5](#_Toc169525166)

[2.3 Entscheiden 5](#_Toc169525167)

[2.4 Realisieren (Implementieren) 5](#_Toc169525168)

[2.4.1 Datenbank erstellen 5](#_Toc169525169)

[2.4.2 Startseite, Login Seite, Registrierseite 5](#_Toc169525170)

[2.4.3 API Methoden Zentral 5](#_Toc169525171)

[2.4.4 Hash Algorithmus 5](#_Toc169525172)

[2.4.5 Token Konfiguration 5](#_Toc169525173)

[2.5 Kontrollieren (Test) 5](#_Toc169525174)

[2.6 Auswerten 5](#_Toc169525175)

# Projektmanagement

## Projektmethodik

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte BeschreibungAls Projektmethode verwende ich das IPERKA Modell. Die klare Struktur dieses Modells hilft bei der Umsetzung des Projekts im Ganzen. Die Informationsbeschaffung wird im 2. Teil des Dokuments klarer beschrieben werden, und ist ebenfalls im Arbeitsjournal ersichtlich. Die Planung im Sinne der Zeit befindet sich im ersten Teil und wird fortgeführt im 2. Teil mit der Planung der Komponenten und Anwendungsstrategien jener Komponenten. Die Restlichen Punkte sind ausschliesslich in der Dokumentation enthalten. IPERKA wird grundsätzlich als Projektmethodik verwendet und auch als Gerüst der Dokumentation. Im Speziellen verwende ich für die Entwicklung das Wasserfallmodell, welches IPERKA sehr ähnelt.

https://konstrukteur-in.ch/wp-content/up 1

Der Arbeitsschritt Realisieren, wird in kleinere Abschnitte unterteilt. Diese werden nacheinander abgearbeitet, dabei verwende ich das Wasserfallmodell als Vorgehensweise. Das heisst es wird immer laufend entwickelt und geplant.

## Zeitplanung

Zeitplanung in Form eines Balkendiagramms mit den Schritten des Wasserfallmodells.

Die Tage sind in 4 Abschnitte unterteilt, 2 am Morgen (8:30-10:35 & 10:50-12:15) und 2 am Nachmittag (13:15-15:00 & 15:10-17:30). Im Diagramm sind die einzelnen Schritte von 1-4 bewertet und entsprechen den tagesabschnitten. Die Dokumentation ist immer Inbegriffen, sowie das Arbeitsjournal.

## Informieren (Analyse)

Ziel des Projekts ist das Bauen eines Logins. Das Login ist nicht einfach normal, sondern es ist auf Sicherheit bedacht. Das bedeutet, nach dem Bauen der Grundstruktur mit Start-, Login- und Registrierungsseite, werden die grössten Sicherheitsprobleme, welche mit dem Login zusammenhängen ausgemerzt. Ziel ist es zu verstehen, was die grössten Lücken heutzutage sind.

Zu den häufigsten Risiken gehören laut Strongdm.com [[1]](#footnote-1)die folgenden:

* **Brute-Force Attack**: Eine Attacke bei der versucht wird das Passwort mit einer Wörterliste herauszufinden, indem man Anfrage um Anfrage schickt.

**Lösung:** die Login Versuche limitieren, 2-Faktor-Atuhentifizierung und starke Passwörter

* **Schwache Credentials**

**Lösung:** beim Login Passwortstärke vorgeben (z.B. Menge an Zeichen, Sonderzeichen, Zahlen usw.)

* **Poor Session Management:** Session identifiers sind nicht geschützt, Sessions können von Angreifern übernommen werden, Sessions werden nicht zerstör bzw. sind immer valid.

**Lösung:** eindeutige Session identifiers, sicherstellen dass Sessions zerstört und gelöscht werden.

* **Staying logged in & Cookie hijacking:** wenn es einen remember me button gibt, können Angreifer das Erstellungsmuster von Cookies erraten und den Account übernehmen mithilfe von XSS (Cross site scripting) und Brute force Attacks.

**Lösung:** cookie verschlüsseln, den Remember me button weglassen.

* **SQL-Injections:** Methode bei der Mithilfe von SQL-Befehlen im Login bestimmte Accounts übernommen werden.

**Lösung:** Parameter basierte Datenbank, Stored Procedures

## Planen (Design)

In der Zeit halte ich mich an den Zeitplan aus dem 1. Teil des Dokuments. Eine Rollenzuweisung gibt es nicht, da ich allein arbeite.

Die Entwicklung wird in kleinere Schritte (siehe 2.4) unterteilt und diese werden mit der Wasserfallmethode bewältigt, d.h. es wird laufend entwickelt getestet usw.

**Datenbankmodell (ERM):**

**Ein Bild, das Text, Schrift, Screenshot, Zahl enthält.

Automatisch generierte Beschreibung**

**UML:**

## Entscheiden

Die Applikation wird in ASP.NET Core umgesetzt und in C# geschrieben. Das liegt daran, dass dieses Framework mir bereits bekannt ist, und mit C# kann ich gut Arbeiten, ausserdem geht das Ganze nach MVC-Architektur. Die API ist eine REST-Schnittstelle und hat einen OData Standard. Die Datenbank ist in MSSQL geschrieben und läuft auf SSMS (SQL Server Management Studio).

Server, Datenbank und API laufen jeweils in einem Docker Container.

Für die Versionierung und Sicherung verwende ich Github.

**Verwendete Packages:**

## Testkonzept

## Arbeitsjournal

# Projektdokumentation

## Realisieren (Implementieren)

### Datenbank erstellen

### Startseite, Login Seite, Registrierseite

### API Methoden Zentral

### Hash Algorithmus

### Token Konfiguration

## Kontrollieren (Test)

## Auswerten

1. <https://www.strongdm.com/blog/authentication-vulnerabilities> [↑](#footnote-ref-1)