Hunter Wilson

IPA Template

IPA-Bericht

Inhaltsverzeichnis

[Teil 1: Umfeld und Ablauf 3](#_Toc165557362)

[1 Aufgabenstellung 3](#_Toc165557363)

[1.1 Ausgangslage 3](#_Toc165557364)

[1.2 Detaillierte Aufgabenstellung 3](#_Toc165557365)

[1.3 Mittel und Methoden 4](#_Toc165557366)

[1.4 Vorkenntnisse 4](#_Toc165557367)

[1.5 Vorarbeiten 4](#_Toc165557368)

[1.6 Neue Lerninhalte 4](#_Toc165557369)

[1.7 Arbeiten in den letzten 6 Monaten 4](#_Toc165557370)

[2 Projektaufbauorganisation 5](#_Toc165557371)

[3 Vorbereitungsarbeit 6](#_Toc165557372)

[4 Firmenstandards 6](#_Toc165557373)

[4.1 Versionsverwaltung 6](#_Toc165557374)

[4.2 Confluence Dokumentation 6](#_Toc165557375)

[4.3 Arbeitsumgebung 6](#_Toc165557376)

[5 Organisation der Arbeitsergebnisse 7](#_Toc165557377)

[5.1 Dokumentation Verwaltung 7](#_Toc165557378)

[5.2 Code Verwaltung 7](#_Toc165557379)

[6 Projektmanagement 7](#_Toc165557380)

[6.1 Anforderungen 7](#_Toc165557381)

[6.2 Gewählte Projektmanagementmethode 7](#_Toc165557382)

[6.3 Alternative Projektmanagementmethoden 8](#_Toc165557383)

[7 Arbeitspakete 9](#_Toc165557384)

[8 Zeitplan 10](#_Toc165557385)

[9 Arbeitsjournal 11](#_Toc165557386)

[9.1 Tag 1 11](#_Toc165557387)

[Teil 2: Projekt 12](#_Toc165557388)

[10 Kurzfassung des Berichts 12](#_Toc165557389)

[11 Analyse 13](#_Toc165557390)

[11.1 Requirement Analysis 13](#_Toc165557391)

[12 Design 13](#_Toc165557392)

[13 Implementation 13](#_Toc165557393)

[14 Testing 13](#_Toc165557394)

[14.1 Testdurchführung 13](#_Toc165557395)

[15 Reflexion 14](#_Toc165557396)

[16 Glossar 14](#_Toc165557397)

[17 Quellenverzeichnis 14](#_Toc165557398)

[Anhang 14](#_Toc165557399)

# Teil 1: Umfeld und Ablauf

## Aufgabenstellung

(Direkt aus die PkOrg.ch Webseite kopiert)

### Ausgangslage

Die Wunderman Thompson Switzerland AG betreibt einen selbstentwickelten Passwort-Manager namens Vault, um private sowie geteilte Accounts und Zugänge zu verwalten. Dieses Tool besitzt sein eigenes Web-Interface. Vault kann nur über das firmeninterne Netzwerk erreicht werden.

Die vorliegende IPA soll die Usability dieses Tools verbessern. Hierzu wird eine Browser Extension für die Browser Google Chrome und Microsoft Edge entwickelt, welche die entsprechenden Accounts und dazugehörigen Passwörter im jeweiligen Kontext anzeigt.

### Detaillierte Aufgabenstellung

Der zurzeit verwendete Passwortmanager wird um eine Browser Extension erweitert, damit die Lösung über die Browser Chrome und Edge hinweg nahtlos verwendet werden kann.

Folgende Elemente der Arbeit bilden den Pflichtteil:   
Entwicklung der Browser Extension:   
Die Erstellung beinhaltet ein vereinfachtes User Interface, welches den nachfolgenden Funktionsumfang erlaubt:  
- Das Nutzen von Passwörtern, welche als Favoriten markiert sind   
- Anbieten einer Auswahl nach URL, Name   
- Einfaches Copy & Paste von Benutzername und Kennwort   
- Unterstützung der Browser Edge (ab Version 110 aufwärts) und Chrome (ab Version 109 aufwärts). Die Versionen aus 2023 oder neuer werden damit abgedeckt.   
- Aufruf des Vault-Eintrages im Web Interface für erweiterte Funktionalität

Die Umsetzung der Benutzerautorisierung via vorgegebener Benutzerrolle und Integrated Security via Kerberos:   
Die Authentisierung wird in den Einstellungen der Extension verwaltet und soll mittels Standardfunktionalität des genutzten Technologiestacks umgesetzt werden (aka Integrated Security), Vorzugsweise mittels Kerberos Delegation der Windows-Benutzersession. (Das Web Frontend unterstützt dies bereits) Es soll nur die Zugangsberechtigungen des jeweiligen User berücksichtigt werden und nur die entsprechenden Einträge an die Browser Extension gesendet werden.

- Integrationskonzept ans existierende Backend (Vault)  
Jegliche API Calls, welche für die Extension verwendet werden, werden im Integrationskonzept schriftlich festgehalten. Dies umfasst: API Call; Verwendungszweck des API Calls; Parameter, welche im Call mitgegeben werden; Antwort des API Calls.

- Umsetzung automatisierbares Testset, welches Integration und Funktionalitätstest umfasst, um die Stabilität der API und sonstige Abhängigkeiten zu gewährleisten. Happy Path Test Cases sind ausreichend. Dieses Testset umfasst mindestens einen Test der folgenden Testarten:   
-Unit Test, welche einzelne Funktionalität der Browser Extension testen   
-Integration Test, welche die Integration von der Browser Extension zu einem gemocketen Backend testen   
-Systemtest, welche die komplette Lösung testen   
Dies dient der Sicherstellung, dass die neue Lösung keine existierende Funktionalität bricht und zukünftige Changes nicht die neue Lösung brechen.

- Die Arbeit wurde entsprechend dem MVC design pattern implementiert und verletzt nicht die clean code Prinzipien.

- Die Zuverlässigkeit der Browserextension wird über eine lückenlose Fehlerbehandlung im ganzen Code konsistent gewährleistet.

### Mittel und Methoden

Für die Entwicklerumgebung wird Visual Studio Code verwendet. Als Programmiersprache kommen HTML, CSS und JavaScript zum Einsatz.

Zur Verfügung stehen dem ausführenden Praktikanten sein Firmenlaptop sowie ein eigener Arbeitsplatz. Ein Austausch mit Fachverantwortlichen ist persönlich oder remote gegeben.

### Vorkenntnisse

Der Praktikant verfügt bereits über Kenntnisse in den Programmiersprachen HTML, CSS und JavaScript.

### Vorarbeiten

Der Praktikant analysiert den bestehenden Passwortmanager "Vault" im Zuge der Vorbereitung. Des weiteren arbeitet sich der Praktikant in die Thematik der Chrome- und Edge-Browserwerweiterungen ein.

### Neue Lerninhalte

Die Thematik vpn Browsererweiterungen für Chrome und Edge ist für den Praktikanten grundsätzlich ein neues Feld. Hierzu gehört auch die gesamthafte Verwaltung inkl. Zuweisung der Passwörter. Dies beinhaltet auch die sichere Übermittlung der Passwörter.

Die Arbeit mit dem Authentifizierungsdienst Kerberos stellt ebenfalls einen neuen und bisher nicht bekannten Bereich dar.

Als eine Quelle wird dem Kandidaten die bereits bestehende Dokumentation zu "Vault" seitens Wunderman Thompson dienen. Ansonsten werden sich die bezogenen Informationen auf öffentlich zugängliche, web-basierte Quellen stützen.

### Arbeiten in den letzten 6 Monaten

Der Lernende bzw. Praktikant hat in den letzten 6 Monaten tatkräftig an Kentico-basierten Kundenprojekten mit C# gearbeitet. Hierzu gehören Arbeiten am Front- sowie Backend und Code. Der Praktikant hat dabei vorrangig an Terminbuchungstools via React-App für verschiedene Kunden gearbeitet und dabei Änderungen am Buchungsprozess in der Test- sowie Liveumgebung durchgeführt. Der Praktikant hat hierbei nicht nur Kundenvorgaben umgesetzt, sondern auch eigene Lösungsvorschläge eingebracht.

Des weiteren hat der Praktikant während eines Monats mittels Godot-Engine eine Schach-Applikation programmiert.

## Projektaufbauorganisation

(Die folgende aufgelistete Einträge sind aus die PkOrg.ch Webseite genommen)

|  |  |
| --- | --- |
| Kandidat | |
| Name | Hunter James Wilson |
| Beschreibung | Führt die IPA aus. |
| Phone | +41 78 401 38 63 |
| Email | hunterinswitzerland@gmail.com |

|  |  |
| --- | --- |
| Verantwortliche Fachkraft | |
| Name | Marco Daniele |
| Beschreibung | Unterstützt die Ausführung der IPA, kann als Hilfeleistung arbeiten bei technischen Schwierigkeiten. Bewertet eventuell diese IPA |
| Phone | 044 448 38 38 |
| Email | Marco.daniele@wundermanthompson.com |

|  |  |
| --- | --- |
| Berufsbildner | |
| Name | Beda Riklin |
| Beschreibung | Unterstützt die Vorbereitung auf den IPA |
| Phone | 044 266 57 57 |
| Email | Beda.riklin@ksh.ch |

|  |  |
| --- | --- |
| Hauptexpert | |
| Name | Janes Thomas |
| Beschreibung | Führt die Expertbesuche aus. Bewertet eventuell diese IPA |
| Phone | 079 295 82 88 |
| Email | janes@janesthomas.ch |

|  |  |
| --- | --- |
| Nebenexpert | |
| Name | Valentin Marolf |
| Beschreibung | Bewertet eventuell diese IPA |
| Phone | +41 79 936 27 80 |
| Email | v.marolf@rafisa.ch |

|  |  |
| --- | --- |
| Betrieb (Durchführungsort) | |
| Name | Wunderman Thompson |
| Adresse | Hardturmstrasse 133 8005 Zürich |
| Phone | 044 448 38 38 |
| Email | Info.ch@wundermanthompson.com |

## Vorbereitungsarbeit

Kandidat hat folgendes vor der IPA gemacht:

* Hat eine Worddokument Template für die Dokumentation erstellt.
* Hat eine Exceldokument Template für das Zeitplan erstellt.
* Hat eine GitHub Repository für die Versionierung von der Dokumentation erstellt.
* Hat eine Kanban Board mit den Kriterien aus dem Kriterienkatalog als Fortschritts-Tracker erstellt.
* Hat seine Arbeitsumgebung für die IPA aufbereitet
* Hat auf developer.chrome.com/docs/extensions verschiedene Informationen über Extensions angeschaut.
* Hat die Wunderman Thompson Vault API und deren Schnittstellen angeschaut.

## Firmenstandards

### Versionsverwaltung

Nach Firmenstandards wird eine Versionsverwaltungstool verwendet, um die Arbeitsergebnisse und die Dokumentation der IPA zu speichern und zu versionieren. Zwei Repositories werden erstellt, eins für die IPA Dokumentation (schon vor der IPA-Start erstellt), und die andere für das Code selbst (Wird währende der IPA erstellt). (Sehe [5.1 Dokumentation Verwaltung](#_Dokumentation_Verwaltung))

### Confluence Dokumentation

Dokumentation zur Installation und Verwendung des Endprodukts von der IPA wird nach Firmenstandards auf die interne Dokumentationstools Confluence geschrieben. Diese Dokumentation soll unter die Vault Page gespeichert. Sämtliche Informationen, die zu dem Dienst vom Endprodukt unterstützen, die aber ausserhalb des Endprodukts selbst liegen, werden auch beschrieben. Ein Link/Kopie dieser IPA-Berichts wird auch auf dieser Seite zur Verfügung gestellt.

### Arbeitsumgebung

Der Kandidat arbeitet während der Ausführung der IPA am Arbeitsplatz beim Betriebsort. Zur Verfügung gestellt wird: eine Windows Arbeitslaptop, die das Flexibilität des Arbeitsort erleichtert. Eine Widescreen Bildschirm, und ein Tisch. Für Mittagspausen hat es eine Kantine im Betrieb, und pro Arbeitswoche werden 2 Tage vom Homeoffice angeboten. Für die Ausführung der IPA hat der Kandidat entschieden die ganze Zeit in Office zu bleiben, um die Fokusse auf der Arbeit zu verbessern.

## Organisation der Arbeitsergebnisse

### Dokumentation Verwaltung

Das IPA-Bericht und alle verwandten Dateien werden mit den Versionsverwaltungstool Git versioniert, und die Repository wird auf die Webseite «GitHub» gespeichert.

Diese Dokumentation wird auf die öffentliche «HW-IPA-Dokumentation» Repository gespeichert (<https://github.com/Hunter-1/HW-IPA-Dokumentation>). Um der täglichen Speicherung der Dokumentation leichter durchsuchbar zu machen, wird am Ende des Tages eine spezielle Release erstellt mit den bis jetzt ausgefühlte Dokumentation darin.

Als Backup wird die Dokumentation auch auf den Betriebs-OneDrive gespeichert, dieser Backup hat leider keine Versionierungsfähigkeit.

### Code Verwaltung

Gleich wie die Dokumentation, wird das Code dieses Projekts mit Git versioniert, und auf GitHub gespeichert. Für das Workflow wird Gitflow verwendet, dabei wird für jedes Arbeitspaket eine eigenes Branch erstellt, bevor es am Masterbranch gemergt wird.

GitHub Repository hier

## Projektmanagement

### Anforderungen

Der Auswahl meine Projektmanagementmethode hat zwei grosse Anforderungen, eins die an diesem Projekt ausrichtet, die andere die an dem Kandidaten ausrichtet. Die Methode soll für eher kleine Projekte geeignet sein, da dieses Projekt allein ausgeführt wird, und nur eine Zeitumfang von 10 Tage hat; und es soll genügend strukturiert sein, weil der Kandidat sich damit gemütlicher fühlt. Was für eine Projektmanagementmethode kann diese zwei Anforderung erfüllen?

### Gewählte Projektmanagementmethode

#### Wasserfallmethode

Die Wasserfallmethode wird für die Projekt gewählt. Diese Arbeitsmethode hat eine lange und bekannte Geschichte in der Arbeitswelt von Softwareentwickler. Heutzutage wird es oft durch andere neue flexibel Arbeitsmethode ersetzt, aber für diese Projekt soll es gut geeignet sein.

Diese Methode ist in 5 Phasen gegliedert:

1. Analysis-Phase
2. Design-Phase
3. Implementation-Phase
4. Testing-Phase
5. Maintenance-Phase

Diese Methode ist für diese Projekt gut geeignet für mehrere Gründen. Die Phasen kommen eins voreinander vor, sie haben eine sehr starke und klare Gliederung der Arbeit, und jede Phase hat eine klare Aufgabe und Zweck. Die Flexibilität von andere Projektmanagementmethoden sind für diese Projekt nicht geeignet, weil die Arbeitsanforderungen aus der IPA alle statisch sind.

Für diese Projekt wird die letzte Phase nicht verwendet, weil Maintenance dieses Projekt ausserhalb der 10 Tage Umfang dieses Projekt liegt. Daher wird diese Phase durch eine Reflexions-Phase ersetzt.

Diese alternative Methode neu gegliedert:

1. Analysis-Phase
2. Design-Phase
3. Implementation-Phase
4. Testing-Phase
5. Reflexions-Phase

### Alternative Projektmanagementmethoden

#### Mini-Projectmanagement (mPM)

Mini Projectmanagement ist besonders gut für kleine Projekte geeignet, weil es in zwei sehr offene Arbeitsphasen unterteilt ist. Die Planungsphase und die Realisierungsphase. Diese Methode ist nicht gewählt worden, weil diese zwei Phasen zu offen für der Kandidat waren, er möchte etwas mit mehr Struktur.

#### IPERKA

IPERKA ist eine oft vorkommende Projektmanagementmethode für Softwareentwicklung. Wie in dem Namen dieser Methode angedeutet, ist IPERKA in 6 Phasen unterteilt:

1. Informieren
2. Planen
3. Entscheiden
4. Realisieren
5. Kontrollieren
6. Auswerten

IPERKA und die ausgewählte Wasserfallmethode haben eine sehr ähnliches Phasen Struktur, diese Methode wird aber nicht ausgewählt, weil der Kandidat eine unklare Verständnis von eine Phase hat, nämlich die Entscheiden-Phase.

## Arbeitspakete

|  |  |  |
| --- | --- | --- |
| **A** | **Umfeld und Ablauf** | |
| A.1 | Allgemeine Inhalt | Allgemeine Information wird im Teil 1: Umfeld und Ablauf geschrieben. Inklusiv: Die Aufgabestellung aus PkOrg, Projektaufbauorganisation, Firmenstandards, Organisation der Arbeitsergebnisse, Projektmanagement |
| A.2 | Zeitplan | Die Arbeitspakete werden definiert und den Zeitplan wird gefüllt |

|  |  |  |
| --- | --- | --- |
| **1** | **Analysis** | |
| 1.1 | Anforderungs-  analyse | Die Anforderungen aus PkOrg werden angeschaut, und eine genaue Liste von Anforderungen werden definiert |
| **2** | **Design** | |
| 2.1 | Ablaufdiagramm | Zeigt den Ablauf einer Normalem Usecase |
| 2.2 | Struktur-diagramm | Zeigt die Struktur vom Programm. Inklusive ist auch das Integrationskonzept ans existierende Backend |
| 2.3 | Testkonzept | Testkonzept wird erstellt |
| **3** | **Implementation** | |
| 3.1 | TODO | Wird, während Design-Phase erstellt |
| **4** | **Testing** | |
| 4.1 | Testausführung | Das Testkonzept wird durchgeführt |
| **5** | **Reflexion** |  |
| 5.1 | Evaluation | Wie nah an die Planung ist das Programm implementiert. Sind alle Anforderung erfühlt, wenn nicht, warum? |
| 5.2 | Kurzfassung | Die Kurzfassung am Anfang vom Teil 2 wird geschrieben |

|  |  |  |
| --- | --- | --- |
| **B** | **Allfällige Dokumentation** | |
| B.1 | Dokumentation | Während das Projekt soll die Implementation gleichzeitig dokumentiert werden. |
| B.2 | Journal | Am Schluss jedes Tages soll in Projekt Journal einen Eintrag geschrieben |
| B.3 | Expert-Visit | Die Hauptexpert besucht an Tag 2 (03.05.24) und an Tag …  Diese Besuche werden richtig protokolliert. |

## Zeitplan

Excel Sheet Goes Here

## Arbeitsjournal

### Tag 1

|  |  |
| --- | --- |
| Tag 1: 02.05.24 | |
| Geplante Arbeitspakete | A.1, A.2, 1.1 |
| Erledigte Arbeitspakete | A.1, A.2, 1.1 |
| Aufgetretene Probleme | - |
| Hilfsmittel | Als Hilfsmittel habe ich alte IPA-Bericht angeschaut und gelesen, um eine gute Gliederung der Texte zu erreichen |
| Vergleich mit Zeitplan | Es gibt keine grossen Abweichungen vom Zeitplan |
| Reflexion | Heute ist alles gut gegangen, Morgen beginne ich mit der richtigen Planung |

# Teil 2: Projekt

## Kurzfassung des Berichts

## Analyse

### Requirement Analysis

Diese Extension soll die Arbeiter bei Wunderman Thompson die Arbeit erleichtern durch das schnelle Aufrufen der gespeicherten Passwörter. Um diese Aufgabe gut zu erfüllen, soll es diese Requirements erfüllen.

#### Functional Requirements

* Das Extension soll eine Liste von allen Passwörtern anbieten, die in Vault als «Favoriten» markiert ist. (Diese Passwörter müssen vorher auf die Vault.wundermanthompson.com Webseite manuell markiert sein).
* Diese Liste soll nach URL oder Name ordnen lassen, diese Ordnung soll wechselbar sein.
* Aus dieser Liste soll es möglich sein, das Password und das Username an den Clipboard zu kopieren (Copy and Paste Fähigkeiten)
* Es soll möglich sein, eine erweiterte Passwort Eintrag in der Extension anzuzeigen, die mehr Information als das Listeansicht hat.
* Das Aufruf von Passwörtern soll die Berechtigungen den Benutzer und seine Benutzerrollen beachten. Unberechtigte Passwörter sollen nicht angezeigt werden

#### Non-Functional Requirements

* Die Extension soll auf Microsoft Edge (ab Version 110) und auf Chrome (ab Version 109) lauffähig sein.
* Das Security des Extensions soll durch Windows Authentication verstärkt.
* Es soll eine umfassende Fehlerbehandlungsmethodik in der Extension eingesetzt
* Da diese Extension für technisch unbewusste Leute ist, soll die Extension so Benutzer freundlich wie möglich sein.

## Design

## Implementation

## Testing

### Testdurchführung

|  |  |
| --- | --- |
| Testfall [Nr.] [Name] | |
| Testperson | Hunter Wilson |
| Testzeitpunkt | [Datum], [Zeit] |
| Beschreibung |  |
| Testumgebung |  |
| Ausgangszustand |  |
| Testdurchführung |  |
| Erwartete Resultat |  |
| Tatsächliche Resultat |  |
| Kommentar |  |
| Status |  |

## Reflexion

## Glossar

## Quellenverzeichnis

# Anhang