

Browsererweiterung auf Chrome und Edge für Vault- Passwortmanager

02.05.2024 – 17.05.2024



Kandidat

Wilson Hunter James

Betrieb (=Durchführungsort)

Wunderman Thompson

Hardturmstrasse 133, PLZ 8005

T +41 78 401 38 63 (am besten erreichbar)

G 044 448 38 38

M hunterinswitzerland@gmail.com

BerufsbildnerIn/ Lehrfirma

Riklin Beda

Kantonsschule Hottingen

Minervastrasse 14, 8032 / Zürich

T 0442665757 (am besten erreichbar)

G 0442665757

M beda.riklin@ksh.ch

Verantwortliche Fachkraft

Daniele Marco

Wunderman Thompson Zurich

Hardturmstrasse 133, 8005 / Zürich

T 044 448 38 38 (am besten erreichbar)

G 044 448 38 38

M marco.daniele@wundermanthompson.com

Hauptexperte

Thomas Janes

T 0792958288 (am besten erreichbar)

G 0792958288

M janes@janesthomas.ch

Arbeitsbereiche

-
-
-
-
-
-
-
-
-

Browsererweiterung auf Chrome und Edge für Vault- Passwortmanager

02.05.2024 – 17.05.2024

Browsererweiterung auf Chrome und Edge für Vault-Passwortmanager

02.05.2024 – 17.05.2024

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.

Browsererweiterung auf Chrome und Edge für Vault-Passwortmanager

02.05.2024 – 17.05.2024

- 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 gemockten 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-Browsererweiterungen ein.

Neue Lerninhalte

Die Thematik von Browsererweiterungen für Chrome und Edge ist für den Praktikanten grundsätzlich ein neues Feld. Hierzu gehört auch die gesamthafte Verwaltung inkl. Zuweisung der

Browsererweiterung auf Chrome und Edge für Vault-Passwortmanager

02.05.2024 – 17.05.2024

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.

Individuelle Kriterien

Auf den folgenden Seiten werden die individuellen Kriterien aufgeführt, welche durch die verantwortliche Fachkraft für diese IPA festgelegt wurden.

Individuelle Kriterien

Leitfrage 1	Grundfunktion des Vault AddIns Die Grundfunktion des Vault AddIns sind funktional vollständig und selbsterklärend umgesetzt: - Einträge zur Auswahl anbieten (nach URL, Name) - Eintrag anwenden (Copy&Paste Benutzername und PW in Loginmaske) - Aufruf des Vault-Eintrages im Web Interface
Gütestufe 3	Alle Grundfunktionen sind funktional und selbsterklärend umgesetzt
Gütestufe 2	Die Grundfunktionen sind vollständig aber nicht selbsterklärend umgesetzt
Gütestufe 1	2 der 3 aufgeführten Grundfunktionen wurden umgesetzt
Gütestufe 0	Mehr als 1 Grundfunktion wurde nicht umgesetzt

Notizen

Individuelle Kriterien

Leitfrage 2	Security
	<p>Ist die Lösung mittels Authentisierung und Autorisierung auf Browser Extension und API Ebene genügend abgesichert?</p> <ul style="list-style-type: none"> - Integrated Security mittels Kerberos Delegation - Berücksichtigung der Zugangsberechtigungen der jeweiligen User - Es werden stets nur berechtigte Einträge angezeigt, an die Browser Extension übermittelt und verarbeitet.
Gütestufe 3	Alle Aspekte wurden vollständig umgesetzt und konzeptionell beschrieben
Gütestufe 2	Alle Aspekte sind konzeptionell vorgesehen und beschrieben aber erst 2 von 3 umgesetzt
Gütestufe 1	Alle Aspekte wurden vollständig umgesetzt aber nicht beschrieben.
Gütestufe 0	Kein Aspekt wurde umgesetzt.

Notizen

Individuelle Kriterien

Leitfrage 3	Codingstyle - lesbarer Code Ist der Code lesbar geschrieben, gut gegliedert und ist die Namensgebung gut gewählt?
Gütestufe 3	Die Namensgebung entspricht den Vorgaben oder ist einfach gut gewählt. Die Struktur des Codes ist ebenfalls gemäss möglicher Richtlinien oder einfach übersichtlich gemacht. Es ist eine gewisse Einheit zu sehen in der Art und Weise, wie der Code strukturiert ist (d.h. es ist überall etwa gleich gemacht).
Gütestufe 2	Die Namensgebung ist ab und zu ungeschickt gewählt, Vorgaben sind teilweise berücksichtigt. Die Codestruktur ist uneinheitlich (so dass Lesbarkeit leidet).
Gütestufe 1	Die Namensgebung ist öfters verwirrend oder unpräzise. Dem Code fehlt es an einigen Stellen an klarer Struktur. Vorgaben sind nicht berücksichtigt.
Gütestufe 0	Die Namensgebung ist verwirrend oder unpräzise. Der Code ist schlecht lesbar.

Notizen

Individuelle Kriterien

Leitfrage 4	Testing
	Stellt ein diverses Testset die Funktionalität sowie Integration der Browser Extension sicher
Gütestufe 3	Es sind folgende Testarten vorhanden: Unittest, welche einzelne Funktionalität der Browser Extension testen Integrationstest, welche die Integration von der Browser Extension zu einem gemocketen Backend testen Systemtest, welche die komplette Lösung testen
Gütestufe 2	Es sind Tests von zwei verschiedenen geforderten Testarten vorhanden.
Gütestufe 1	Es sind nur Tests einer geforderten Testart vorhanden.
Gütestufe 0	Es sind nur Tests vorhanden, welcher keiner der drei geforderten Testarten entspricht.

Notizen

Individuelle Kriterien

Leitfrage 5	MVC (Programmierung)
	Ist die Auftrennung nach dem MVC-Pattern konsequent durchgeführt und sind Abweichungen vom Pattern beschrieben und begründet?
Gütestufe 3	M: konsequent nur Datenaufbereitung für GUI V: konsequent nur Darstellung (GUI) C: konsequent nur Ablaufsteuerung und Validierung Sind die Firmenstandards bzw. Firmenusanzen betreffend Realisierung eingehalten? Sind die Schnittstellen im Code klar ersichtlich? Alle 5 Aspekte erfüllt
Gütestufe 2	4 Aspekte erfüllt
Gütestufe 1	3 Aspekte erfüllt
Gütestufe 0	weniger als 3 Aspekte erfüllt

Notizen

Individuelle Kriterien

Leitfrage 6	Integrationskonzept ans existierende Backend
	Werden die verwendete API Calls nachvollziehend dokumentiert und deren Verwendung begründet.
Gütestufe 3	Alle verwendeten API Calls sind vollständig aufgelistet und vollständig dokumentiert.
Gütestufe 2	Alle verwendeten API Calls sind aufgelistet, die Beschreibung ist jedoch unvollständig.
Gütestufe 1	Die verwendeten API Calls sind nicht vollständig aufgelistet und/oder keine Beschreibung ist dazu vorhanden.
Gütestufe 0	Keine Dokumentation oder Erklärung der verwendeten API Calls

Notizen

Leitfrage 7	Codierung: Fehlerbehandlung Fehlerbehandlung: Werden mögliche Fehler mit den entsprechenden Mitteln erkannt und behandelt?
Gütestufe 3	Es wurde eine Strategie zur Fehlerbehandlung eingeführt und im ganzen Code konsistent verwendet.
Gütestufe 2	Die Fehlerbehandlung ist lückenhaft und/oder die Fehler bleiben nach der Erkennung unbehandelt.
Gütestufe 1	Die Fehlerbehandlung wurde oft vergessen. Code streckenweise ohne Fehlertests (wo sie als notwendig erachtet würden).
Gütestufe 0	Fehlerbehandlung nicht oder nur sehr rudimentär vorhanden.
