



## Semesterarbeit Einreichungsformular

### Klasse ITCNE24 – 3. Semesterarbeit

Im Folgenden beschreibt der Studierende das geplante Thema seiner Zertifikatsarbeit. Der Studierende orientiert sich dabei an dem Bewertungsraaster. Der begleitende Dozent entscheidet, ob es sich dabei um ein geeignetes Thema handelt und fügt seine Kommentare und Überlegungen hinzu.

Bitte die *kursiven Textteile* durch die konkreten Angaben ersetzen.

#### Name und Vorname des Studierenden

Miguel Schneider

#### Titel der Semesterarbeit

Lizenzüberwachungstool für die ISE AG

#### Zu behandelnde Themenfelder / Module (bitte ankreuzen)

Pkt.	Themenfeld / Modul	
1.	Projektmanagement (Corrado Parisi)	Pflicht
2.	Relationale Datenbanken (Yves Nussle)	<input type="checkbox"/>
3.	NoSQL, Key/Value, Time Series (Yves Nussle)	<input type="checkbox"/>
4.	Microservices mit Python (Boris Langer)	X
5.	Datensicherheit / Datenschutz (Roman Stammbach)	<input type="checkbox"/>
6.	Netzwerk (Philipp Albrecht)	<input type="checkbox"/>
	<b>Optional: Thema aus Semester 2</b>	<input type="checkbox"/>
7.	<u>Netzwerktechnik / Netzwerkarchitektur</u> (Philipp Albrecht)	<input type="checkbox"/>
8.	Infrastructure as Code (Ansible, CI/CD Pipelines) (Armin Dörzbach)	<input type="checkbox"/>
9.	Geschäftsprozesse / BPMN / Workflow (Caesar Roth)	<input type="checkbox"/>
10.	IT Service Management (SLA, Service Katalog) (Armin Dörzbach)	<input type="checkbox"/>
11.	FAAS – Functions as a Service (Python, FAAS in der Cloud) (Armin Dörzbach)	<input type="checkbox"/>

## **Kurzbeschreibung des Zertifikatsarbeit Themas (1 bis max. 2 Seiten)**

### **Problemstellung / Ausgangslage / Potential der Semesterarbeit**

Die ISE AG betreut als IT-Dienstleister eine Vielzahl an Firmen und Schulen, wobei eine zuverlässige Lizenzüberwachung essenziell ist. Momentan werden Lizenzprobleme oft erst erkannt, wenn Kunden bereits eine Störung melden.

Dieses Projekt hat das Ziel, ein zentrales Tool zu schaffen, das frühzeitig auf potenzielle Lizenzengpässe hinweist – automatisiert, übersichtlich und intern nutzbar.

#### **Technischer Aufbau:**

Das Lizenzüberwachungstool basiert auf einem Flask-Microservice, der regelmässig über die Microsoft Graph API Lizenzinformationen aus den Kundensystemen abrufen. Die gesammelten Daten werden zentral in einer SharePoint-Liste abgelegt. Der Einsatz von SharePoint als Datenspeicher wurde gewählt, da die ISE AG bereits intensiv mit dieser Plattform arbeitet – was eine einfache Integration, Berechtigungssteuerung und Weiterverarbeitung (z. B. mit Power Automate) ermöglicht.

Der Microservice stellt zusätzlich eine API für das Frontend bereit. Das Frontend zeigt die Lizenzdaten in strukturierter und übersichtlicher Form an und unterstützt Mitarbeitende dabei, mögliche Engpässe frühzeitig zu erkennen.

Zur Frühwarnung bei kritischen Lizenzständen wird die SharePoint-Liste von Power Automate überwacht. Bei definierten Schwellenwerten (z. B. ablaufende Lizenzen) werden automatisch Benachrichtigungen an das zuständige Team versendet.

### **Zielsetzung der Semesterarbeit.**

1. **Automatisierte Lizenzdatenerfassung**  
Entwicklung eines Microservices, der Lizenzinformationen selbstständig aus Kundensystemen via Microsoft Graph API abrufen.
2. **Zentrale Datenhaltung**  
Speicherung der gesammelten Lizenzdaten in einer SharePoint-Liste zur einfachen Integration in die bestehende ISE-Umgebung.
3. **Visuelle Darstellung im Frontend**  
Bereitstellung eines benutzerfreundlichen Frontends, das eine schnelle und übersichtliche Einsicht in den Lizenzstatus ermöglicht.
4. **Automatisierte Benachrichtigungen**  
Implementierung einer automatischen Alarmierung bei kritischen Zuständen mittels Power Automate.
5. **Mehrwert für den Betrieb schaffen**  
Schaffung eines Tools, das real im ISE-Alltag eingesetzt werden kann und Effizienz sowie Qualität im Lizenzmanagement erhöht.

## Terminplan mit den wesentlichen Arbeitsschritten

Datum	Aktivität	Wer	Empfänger
26.03.25	Ablauf Semesterarbeiten und Liste mit Projektthemen vorstellen	Lehrgangsleitung	Studierende
16.04.25	Abgabe Einreichungsformular	Studierende	Experte/innen
05.05.25	Freigabe Semesterarbeit durch Expert/innen	Experte/innen	Studierende
05.05.25	Beginn Umsetzung Semesterarbeit	Studierende	Experte/innen
09.05.25	1. Sprint, evtl. mit Besprechung	Studierende	Experte/innen
02.06.25	2. Sprint, evtl. mit Besprechung	Studierende	Experte/innen
20.06.25	3. Sprint, evtl. mit Besprechung	Studierende	Experte/innen
09.07.25	Abgabe der Arbeit / Abnahme mit Schlusspräsentation	Studierende	Experte/innen
18.07.25	Notenvorschlag	Projekt-Experte/in	Lehrgangsleitung
25.07.25	Mitteilung der Noten mit individuellem Feedback zur Einsicht hochladen	Projekt-Experte/in	Studierende
11.08.25	Mitteilung der Noten	Lehrgangsleitung / Sekretariat	Studierende

## Sachmittel / Rahmenbedingungen

Für die erfolgreiche Umsetzung der Semesterarbeit und zur Erstellung der Projektergebnisse werden folgende Sachmittel und Rahmenbedingungen benötigt:

- 1. Docker Container (Docker Desktop)**  
Zur lokalen Entwicklung und Ausführung des Flask-Microservices in einer isolierten und kontrollierten Umgebung.
- 2. MSVC-Flaskservice (vorbereitete Grundstruktur aus dem Unterricht)**  
Als technische Grundlage dient eine im Unterricht vorbereitete Flask-Service-Struktur, auf der die Schnittstellenanbindung und die Logik des Projekts aufgebaut werden.
- 3. GitHub / GitHub Pages (für Dokumentation)**  
GitHub wird für die Quellcode-Verwaltung sowie zur Bereitstellung der Projektdokumentation über GitHub Pages verwendet. Ein Deployment erfolgt nicht über GitHub, da die Anwendung zunächst nur in einer lokalen Testumgebung betrieben wird.
- 4. Microsoft 365 Zugang (Test-Tenant)**  
Die Lizenzdaten werden aus einem dedizierten **Test-Tenant** abgerufen, der bereits in der zweiten Semesterarbeit verwendet wurde. Dieser Tenant stellt alle nötigen Microsoft 365 Ressourcen sowie die Berechtigungen für die Graph API zur Verfügung.
- 5. SharePoint Online Umgebung (innerhalb des Test-Tenants)**  
Innerhalb desselben Test-Tenants steht eine SharePoint Online Instanz zur Verfügung, in der die Lizenzinformationen gespeichert und später durch Power Automate verarbeitet werden.

## Vorgaben, Methoden und Werkzeuge

Im Rahmen dieses Projekts werden definierte Methoden, Werkzeuge und technische Vorgaben eingesetzt, um eine strukturierte, praxisnahe und zielgerichtete Umsetzung sicherzustellen.

### Methoden:

- **Scrumban** – Agile Aufgabenplanung mit visueller Darstellung und laufender Priorisierung
- **Lean** – Fokus auf schlanke, effiziente Prozesse und kontinuierliche Verbesserung
- **Modularisierung** – Trennung der Anwendung in eigenständige, wartbare Komponenten

### Werkzeuge:

- **Visual Studio Code** – Hauptentwicklungsumgebung für Frontend und Backend
- **Docker Desktop** – Containerisierung zur lokalen Ausführung des Flask-Microservices
- **GitHub & GitHub Projects** – Versionsverwaltung des Quellcodes, Dokumentation und visuelle Aufgabenorganisation im Rahmen des Scrumban-Prozesses
- **Microsoft 365 (Graph API)** – Lizenzabfragen aus dem dedizierten Test-Tenant
- **SharePoint Online & Power Automate** – Speicherung und automatisierte Auswertung der Lizenzdaten

### Vorgaben:

- Nutzung eines bestehenden Microsoft 365 Test-Tenants (aus vorangegangener Semesterarbeit)
- Verwendung des im Unterricht bereitgestellten Flask-Grundgerüsts
- Dokumentation und Fortschrittsverfolgung erfolgen über GitHub

## Risiken

Risiko	Eintritt	Auswirkung	Massnahme zur Vermeidung / Minderung
<b>Zeitverzögerung durch technisches Debugging oder fehlendes Know-how</b>	<b>Hoch</b>	Mittel	Zeitpuffer einplanen, Fokus auf MVP, Probleme frühzeitig angehen
<b>Fehlerhafte oder unvollständige API-Abfragen (Graph API)</b>	<b>Mittel</b>	Hoch	Ausgiebig mit Testdaten prüfen, Logging & Fehlerbehandlung implementieren
<b>SharePoint-Zugriffsprobleme (z. B. Berechtigungen, API-Limits)</b>	<b>Mittel</b>	Mittel	Rechte und Zugriff frühzeitig testen, Alternativlösung vorbereiten
<b>Power Automate Benachrichtigungen funktionieren nicht zuverlässig</b>	<b>Niedrig</b>	Hoch	Flows früh einrichten, mit Testfällen absichern, manuelle Kontrolle ergänzen
<b>GitHub-Dokumentation wird nicht laufend gepflegt</b>	<b>Niedrig</b>	Niedrig	Doku fix in Workflow einplanen, regelmässige Erinnerung im Taskboard

## Entscheid des begleitenden Dozenten

Bitte ankreuzen

- Genehmigt
- Zu verbessern
- Abgelehnt

## Begleitender Dozent

Ort und Datum: \_\_\_\_\_

Name & Unterschrift: \_\_\_\_\_

## Beurteilung des Antrages

Kriterium	Kommentare	Erfüllt
<b>Anforderungen an die Form (Strukturierung) der Semesterarbeit</b>		
Problemstellung		<input type="checkbox"/>
Ziele (mindestens drei!)		<input type="checkbox"/>
Risiken bezogen auf Zertifikatsarbeit		<input type="checkbox"/>
Themenfelder (mindestens zwei!)		<input type="checkbox"/>
<b>Anforderungen an Qualität der Semesterarbeit</b>		
Machbar		<input type="checkbox"/>
Praxisnah		<input type="checkbox"/>
Herausfordernd		<input type="checkbox"/>
Lehrgangsbezug		<input type="checkbox"/>

Damit die Semesterarbeit angenommen wird, müssen alle Kriterien erfüllt sein.

Die Kommunikation wird über den jeweiligen Teams-Kanal geführt.